Markov modeling, discrete-event simulation – Module B

5XIE0 Computational Modeling

Twan Basten, Marc Geilen, Jeroen Voeten
Electronic Systems Group, Department of Electrical Engineering

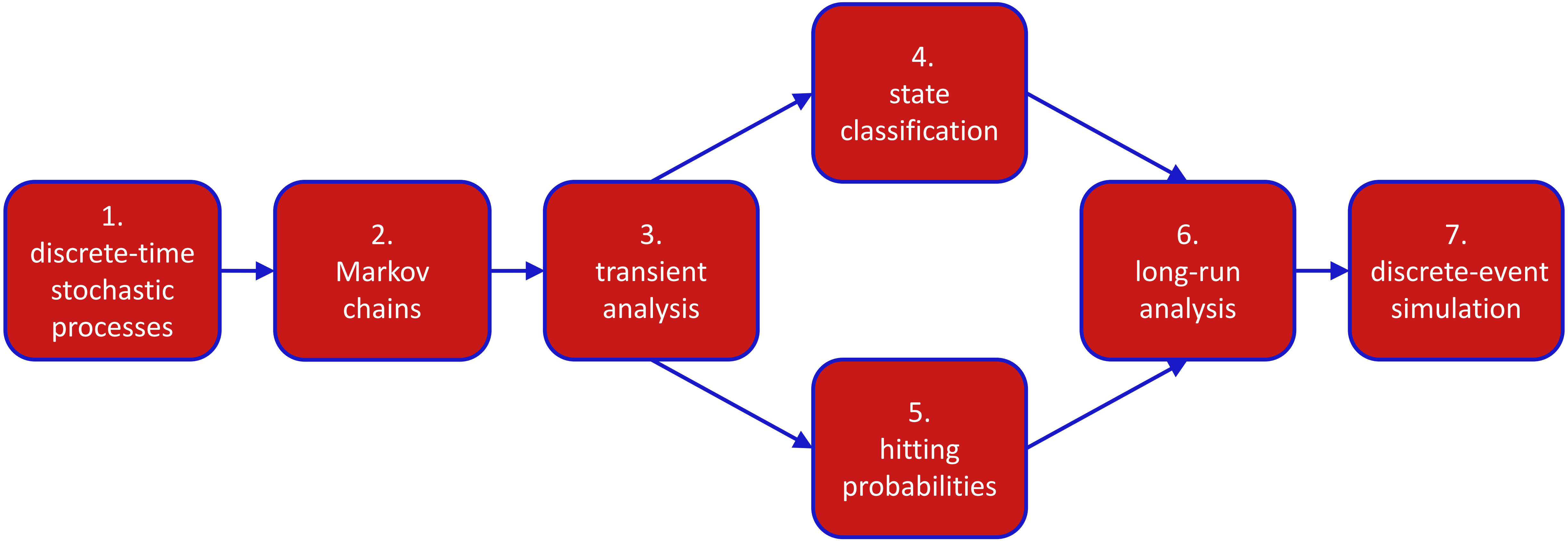
credits and sources

- wikipedia:
  - https://en.wikipedia.org/wiki/Markov_chain
- Seminal textbooks on probability theory and Markov theory:
  - A.F. Karr. Probability. Springer, 1993.
  - N. Privault. Understanding Markov Chains, Examples and Applications. Springer, 2018.
  - J.R. Norris. Markov Chains. Cambridge University Press, 2012.
  - D.R. Cox and H.D. Miller. The Theory of Stochastic Processes. Springer, 1967.
  - K.L. Chung. Markov Chains with Stationary Transition Probabilities. Springer, 1967.

why, what, how

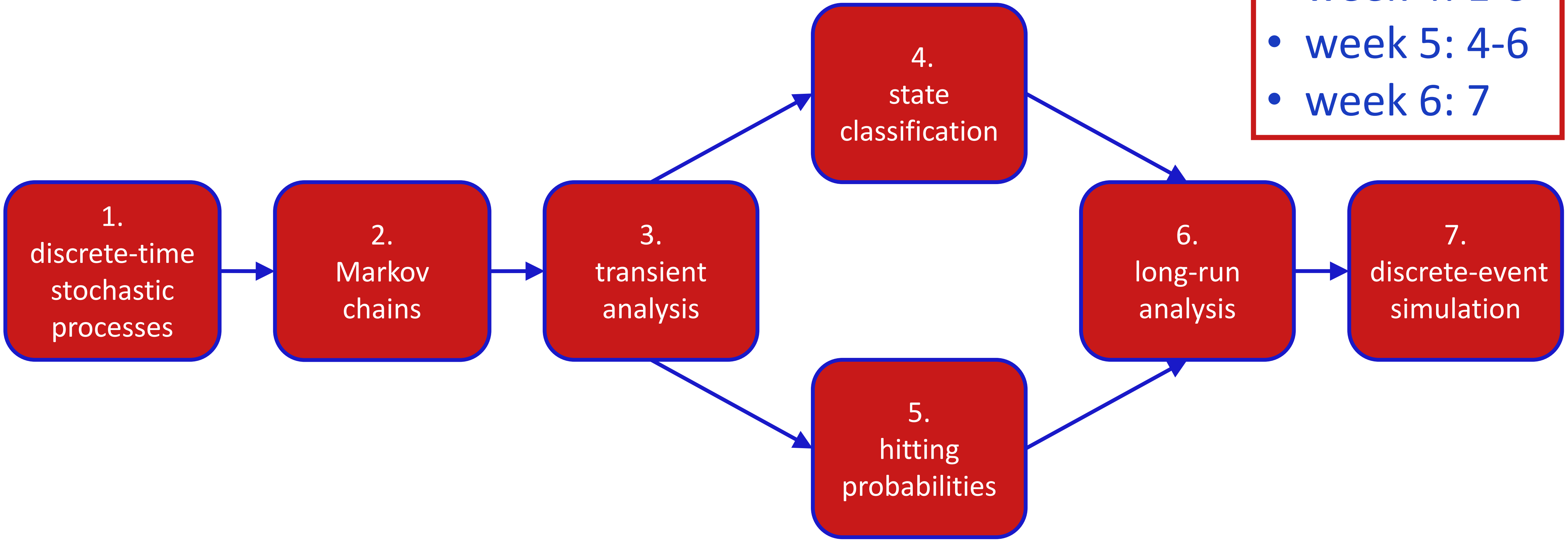
- *why*: get insight in systems that behave **randomly**
  1. gambler playing a game of roulette in the casino
  2. autonomous rover trying to get its way out of a maze
  3. Internet forwarding packets that are offered at random times
  4. video application displaying a movie streamed via the Internet
- *what*: **model** these systems as *Markov chains*
- *how*: **analyze** the properties of interest
  1. probability that the gambler is broke after 10 spins of roulette wheel
  2. expected time before the rover has found the exit of the maze
  3. expected number of packets the Internet can forward per second (throughput)
  4. expected fraction of time that a hiccup occurs

module B - submodules and dependencies



module B - submodules and dependencies - schedule

- week 4: 1-3
- week 5: 4-6
- week 6: 7



course material and exercises

- course notes ‘Computational Modeling’, part B
  - sections correspond to submodules of module B

- each submodule / section accompanied by several exercises
  - answers to all exercises are in Section B.8 of the course notes
  - CMWB can be used for many exercises

B.1 - discrete-time stochastic processes

discrete-time stochastic process

- a **discrete-time stochastic process** is a **sequence** of **random variables** $X_0, X_1, \cdots$

- X_n $(n = 0,1, \cdots)$ models the **random state** of a system at **time** n where
  - n can refer to an actual moment in time, but can also denote an index in a sequence of events
  - X_0 is the **initial state**
  - X_n assumes values in a finite set $S = \{1, 2, \cdots, N\}$ of **states** called the state-space

- model evolution over time of systems or phenomena with a random character, e.g. X_n can denote
  - the number of packets in a buffer at time n
  - the waiting time of the nth person arriving in a queue

probability distribution

- $P(X_n = i)$ denotes the **probability** that the system is in state i at time n
  - where $i \in S$ is called a **realization** of X_n

- $\pi^{(n)}: S \rightarrow [0,1]$ is the **probability distribution** of X_n
  - where $\pi^{(n)}(i) = P(X_n = i)$

- $\pi^{(n)}$ is considered to be **row vector** $[\pi_1^{(n)}, \pi_2^{(n)}, \cdots, \pi_N^{(n)}]$ (because of shape of S)
  - so $\pi_i^{(n)}$ is the probability that the system is in state i at time n
  - $\sum_{i=1}^{N} \pi_i^{(n)} = 1$

rewards

- a function $r: S \rightarrow \mathcal{R}$ is called a **reward**
  - random variable $r(X_n)$ denotes the reward that is earned at time n
  - $R = \{r(i) \mid i \in S\}$ is the set of possible realization of $r(X_n)$

- $P(r(X_n) = v) = \sum_{i \in S, r(i)=v} \pi_i^{(n)}$
  - the probability that reward v ($v \in R$) is earned at time n

- $E\big(r(X_n)\big) = \sum_{v \in R} v\, P(r(X_n) = v) = \pi^{(n)} \cdot r^T$
  - the **expected reward** earned at time n
  - r is considered to be **row vector** [r(1), r(2), ⋯, r(N)]

example – game of dice

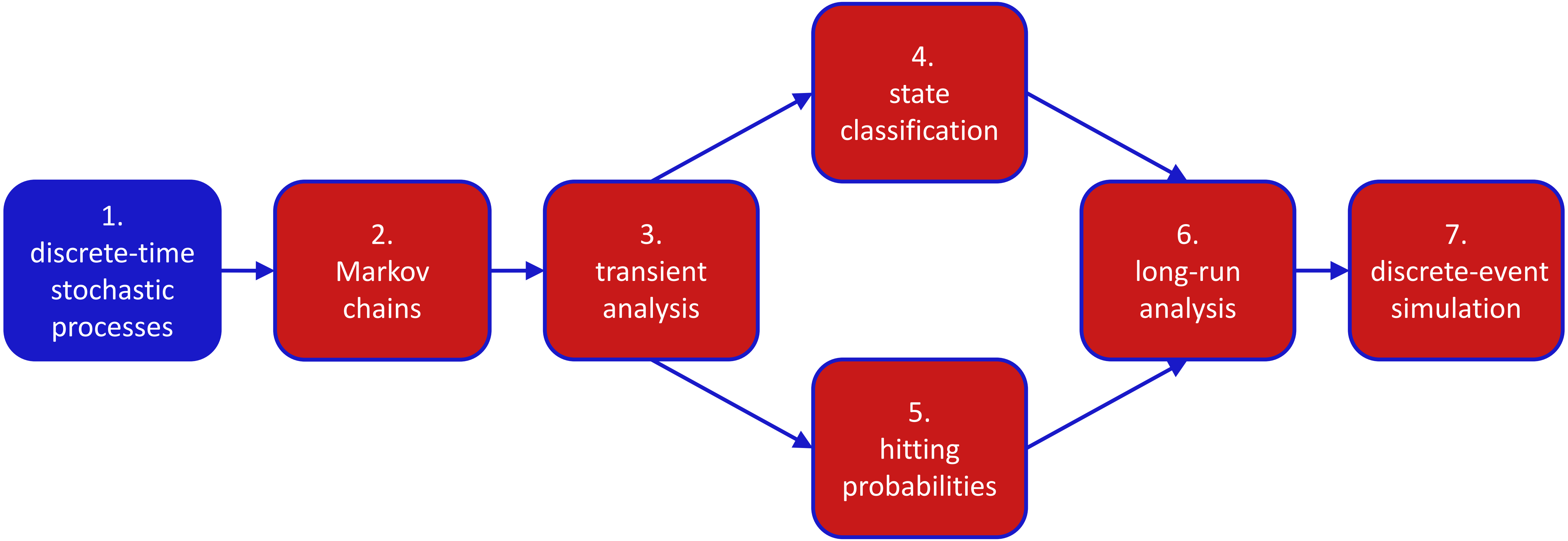
- player repeatedly tosses a die
  - pay €1 in case of 1, 2, 3
  - earn €1 in case of 4, 5, 6

- model as discrete-time stochastic process $X_0, X_1, \cdots$
  - X_{n-1} models random outcome of n^{th} toss
  - state-space $S = \{1, 2, 3, 4, 5, 6\}$
  - distribution $\pi^{(n)} = \left[\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}\right]$
  - reward $r = [-1, -1, -1, 1, 1, 1]$
  - expected reward $\pi^{(n)} \cdot r^T = 0$

discrete-time stochastic processes – exercises

- Section B.1 in the course notes
    - Exercise B.1 (Wealthy gambler – expected reward)
    - Exercise B.2 (Time-slotted Ethernetwork – throughput)
    - Exercise B.3 (Expected reward – computation)
- answers are provided in Section B.8 of the course notes

module B - submodules and dependencies



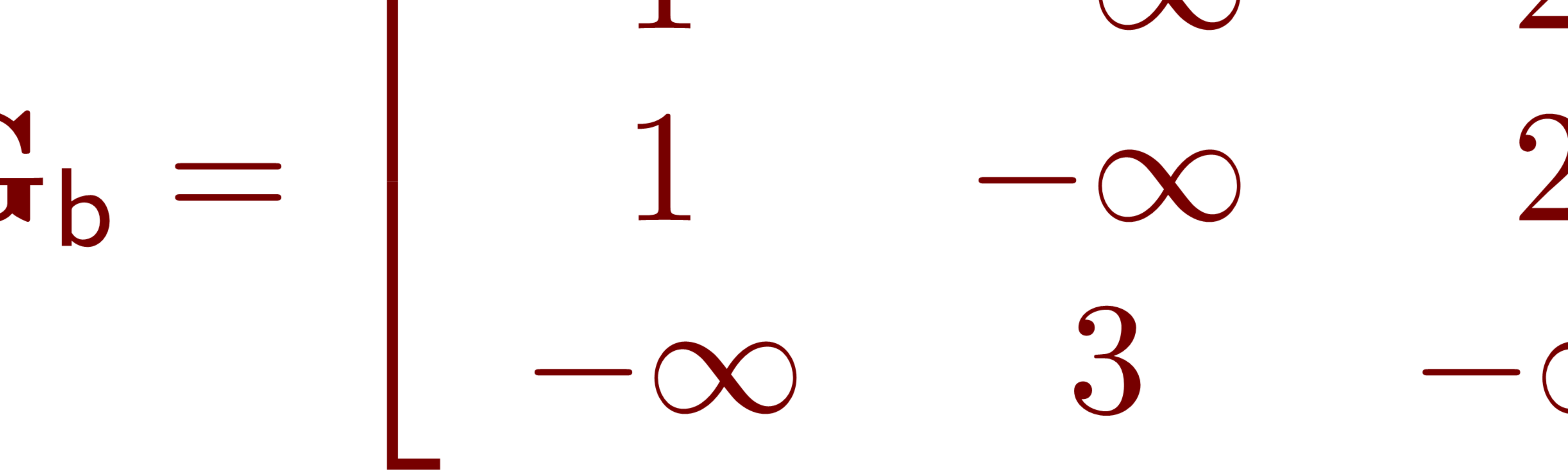
lesson learned

- a **discrete-time stochastic process** is a sequence of random variables $X_0, X_1, \cdots$
  - where X_n denotes the **state** at **time** n
  - where X_n assumes values in **state-space** $S = \{1, 2, \cdots, N\}$
  - $\pi^{(n)}$ denotes the **probability distribution** vector of X_n

- a **reward** r is a real-valued row vector [r(1), r(2), ⋯, r(N)]
  - **expected reward** at time n is given by $\pi^{(n)} \cdot r^T$

questions

- can we always compute probability distributions and expected rewards?



B.2 – Markov chains

independent identically distributed variables (1)

- the random variables of process $X_0, X_1, \cdots$ are
  - **identically distributed** iff $\pi^{(m)} = \pi^{(n)}$ for all n, m
  - (pairwise) **independent** iff $P(X_m = i, X_n = j) = P(X_m = i) \cdot P(X_n = j)$ for all $n \neq m$ and $i, j \in S$
  - (pairwise) **independent** iff $P(X_n = i \mid X_m = j) = P(X_n = i)$ for all $n \neq m$ and $i, j \in S$

- $P(X_n = i \mid X_m = j)$ denotes the probability that the process is in state i at time n *given the fact* that it is in state j at time m
  - $P(X_n = i \mid X_m = j) = \frac{\mathrm{P(X_n=i,X_m=j)}}{P(X_m=j)}$

- example: player repeatedly tossing a die yields a process with independent identically distributed variables
  - distribution $\pi^{(n)} = \left[\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}\right]$

identically distributed random variables (2)

- advantage: relatively easy to analyze
  - if distribution is known for some X_m it is known for all X_n
  - expected rewards $E(r(X_m))$ can be readily computed
  - application of powerful theorems such as central limit theorem and strong law of large numbers

- disadvantage: many systems cannot be modeled
  - e.g. amount of cash of roulette player after n^{th} spin
  - e.g. number of packets in packet buffer at time n

Markov chains (1)

- Markov chains
  - after influential Russian mathematician Andrey Markov
  - rich enough to capture dependencies between variables
  - sufficient structure to support analysis of interesting properties

- Broad range of applications
  - performance analysis of embedded systems and communication systems
  - stochastic software testing
  - data compression
  - page ranking for websites
  - insurance risk management
  - Biomedical engineering
  - …

Markov chains (2)

- a **Markov chain** is a discrete-time stochastic process $X_0, X_1, \cdots$ such that
  - $P(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, \cdots, X_0 = i_0) = P(X_{n+1} = j \mid X_n = i)$
  - for all $n \geq 0$ and $i, j, i_0, \cdots i_{n-1} \in S$

- Markovian property: the random state at time *n+1* is determined completely by the state realized at time *n*, and is not dependent on the states visited before *n*

- a Markov chain is assumed to be **time homogeneous**
  - for each pair $i, j \in S$, a constant P_{ij} exists such that for all $n \geq 0$, $P(X_{n+1} = j \mid X_n = i) = P_{ij}$

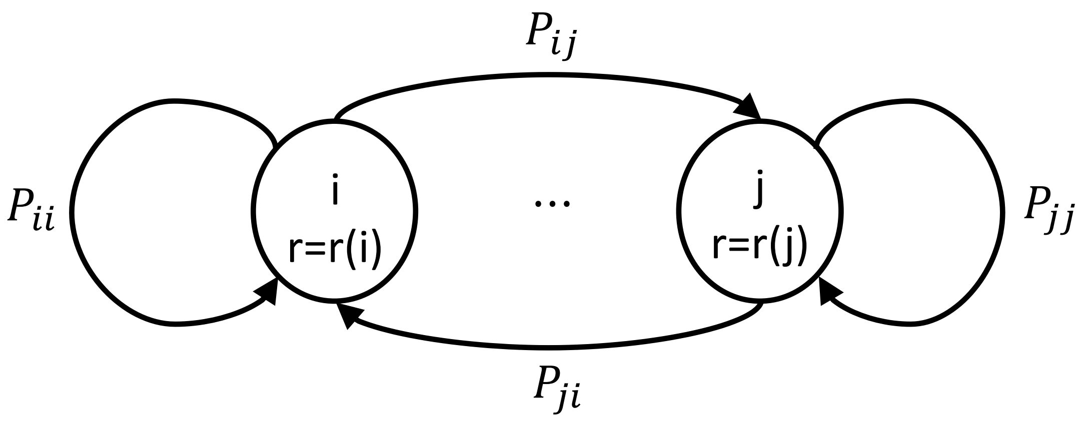
transition probability matrix

- $P(X_{n+1} = j \mid X_n = i) = P_{ij}$
  - is the probability that state *j* is visited at time *n+1* given the system is in state *i* at time *n*
  - is the probability that the system transits from state *i* to state *j* in a single step
  - is called **one-step transition probability**

- organized as square **transition probability matrix** *P*
  - $P = \begin{bmatrix} P_{11} & \cdots & P_{1N} \\ \vdots & \ddots & \vdots \\ P_{N1} & \cdots & P_{NN} \end{bmatrix}$
  - P_{ij} positioned in row *i*, column *j*
  - $\sum_{j \in S} P_{ij} = 1$ for each $i \in S$

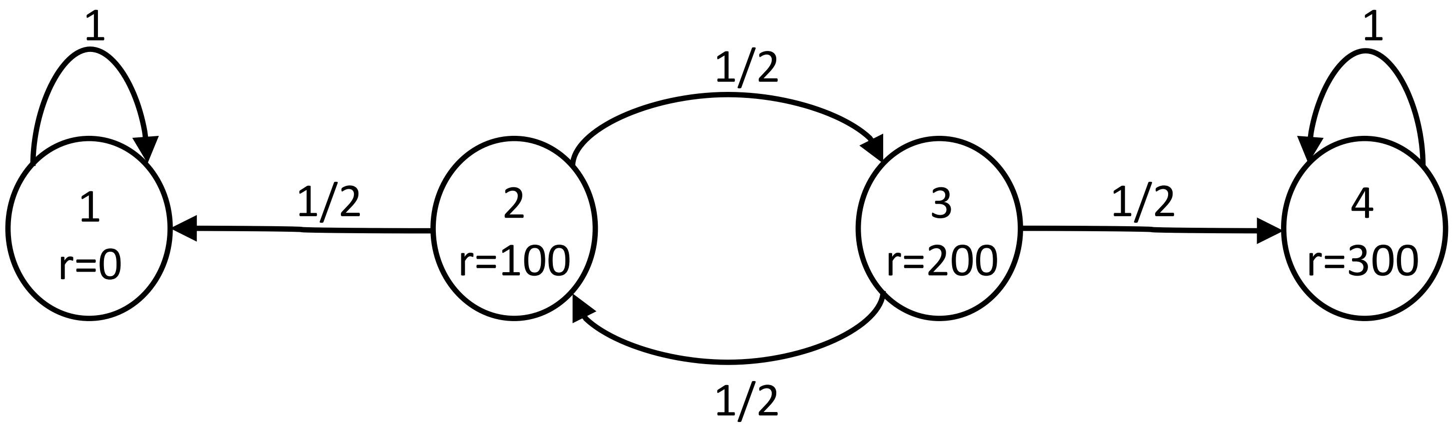
transition diagram

- graphical representation of transition probability matrix
  - *N nodes*, labeled $1 \cdots N$
  - directed *arc* is drawn between node i and node j iff $P_{ij} > 0$
  - optional in case reward *r* is defined: mark node *i* with *r=r(i)*



example – gambler's ruin

- gambler playing roulette
  - initial capital €100
  - bet €100 on black at each spin
  - when black is hit, earn €100
  - when red is hit, lose €100
  - quite when either €0 or €300 cash



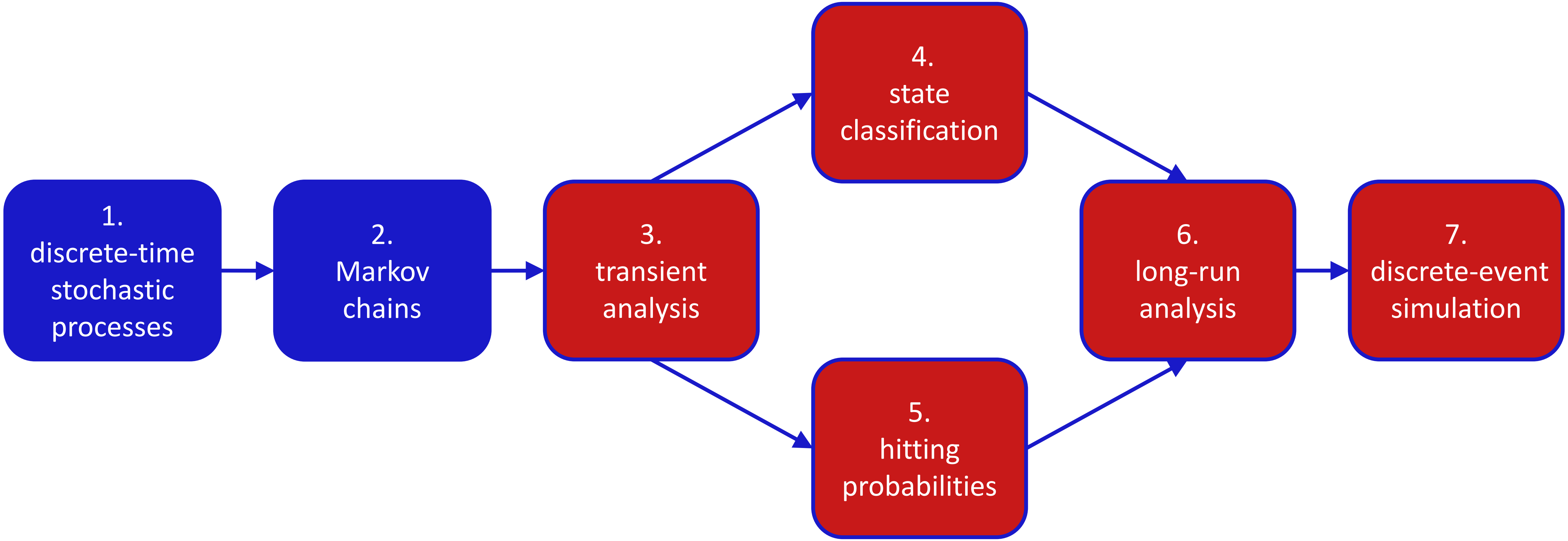
- Markov chain
  - state space $S = \{1, 2, 3, 4\}$
  - reward $r: S \rightarrow \mathcal{R}, r(1) = 0, r(2) = 100, r(3) = 200, r(4) = 300$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Markov chains – exercises

- Section B.2 in the course notes
  - Exercise B.4 (Transition diagram to matrix)
  - Exercise B.5 (Matrix to transition diagram)
  - Exercise B.6 (Markov chains – dependent and non-identically distributed variables)
  - Exercise B.7 (Gambler’s ruin – probability distributions)
    - use CMWB (DTMC) to double check your answers
      1. select ‘Create a new DTCM model’ in ‘General Operations’, enter the Gambler’s ruin model and save
      2. select ‘View Transition Diagram’ in ‘Operations on Markov chains’ to inspect the transition diagram
      3. selected ‘Transient Distribution’ and enter a number (say 2) of steps to analyze
      4. distribution vectors are provided in ‘Analysis Output’ pane
  - Exercise B.8 (Markov chains – independent identically distributed variables)
    - use CMWB (DTMC) to double check your answers to (a)
      1. select ‘Create a new DTCM model’ in ‘General Operations’ enter the model and save
      2. select ‘Transient Distribution’ and enter the number of steps to analyze
- answers are provided in Section B.8 of the course notes

module B - submodules and dependencies



lesson learned

- the variables in discrete-time stochastic process are **independent identically distributed** if they are **mutually independent and have the same probability distribution**
- a **Markov chain** is a discrete-time stochastic process that satisfies the **Markovian property** and is **time-homogeneous**
  - the random state at time $n+1$ is determined completely by the state realized at time n
  - the transition probabilities are independent of the time
- Markov chains are represented by **transition probability matrices** or **transition diagrams**

questions

- can we always compute probability distributions and expected rewards?

questions

- can we always compute probability distributions and expected rewards? no, but we can do so for sequences of independent identically distributed variables and for Markov chains
- how to compute the probability vector (and expected reward) at time n?
- how to compute the probability that the Markov chain transits from state i to state j in n steps?
- can we also compute the limiting behavior when n grows very large?

B.3 – transient analysis

transient analysis: compute probability distributions

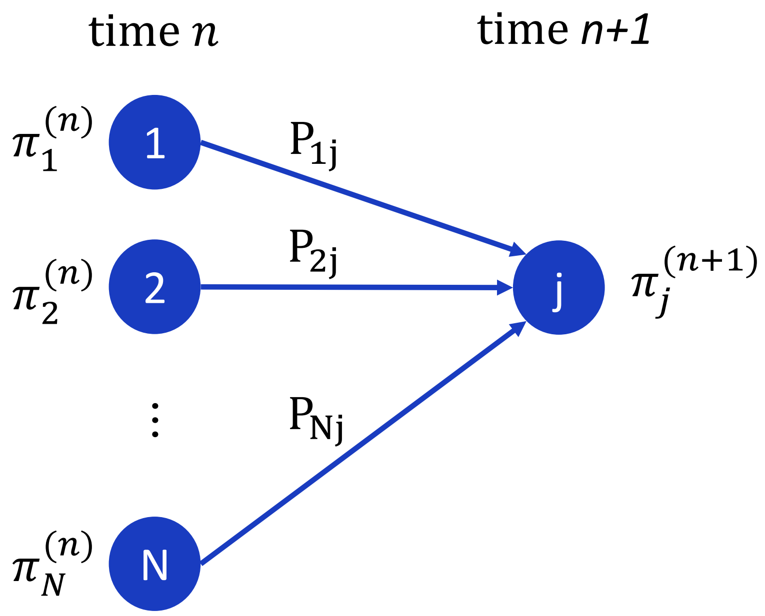
- Recall
  - Markov chain variables $X_0, X_1, \cdots$ have probability vectors $\pi^{(0)}, \pi^{(1)}, \cdots$
  - $\pi_i^{(n)} = P(X_n = i)$: the probability that the system is in state i at time n
  - $\pi^{(0)}, \pi^{(1)}, \cdots$

- Question
  - given initial distribution $\pi^{(0)}$
  - can we compute $\pi^{(n)}$ for each $n = 0,1, \cdots$

law of total probability

- **law of total probability**
  - $P(X_m = j) = \sum_{i \in S} P(X_m = j, X_n = i) = \sum_{i \in S} P(X_m = j \mid X_n = i)\, P(X_n = i)$
  - for all $n, m = 0,1,\cdots$ and $i, j \in S$

- in particular
  - $P(X_{n+1} = j) = \sum_{i \in S} P(X_{n+1} = j \mid X_n = i)\, P(X_n = i)$
  - thus $\pi_j^{(n+1)} = \sum_{i \in S} P_{ij}\, \pi_i^{(n)} = \sum_{i \in S} \pi_i^{(n)} P_{ij}$
  - so $\boldsymbol{\pi_j^{(n+1)} = \pi_1^{(n)} \cdot P_{1j} + \pi_2^{(n)} \cdot P_{2j} + \cdots + \pi_N^{(n)} \cdot P_{Nj}}$



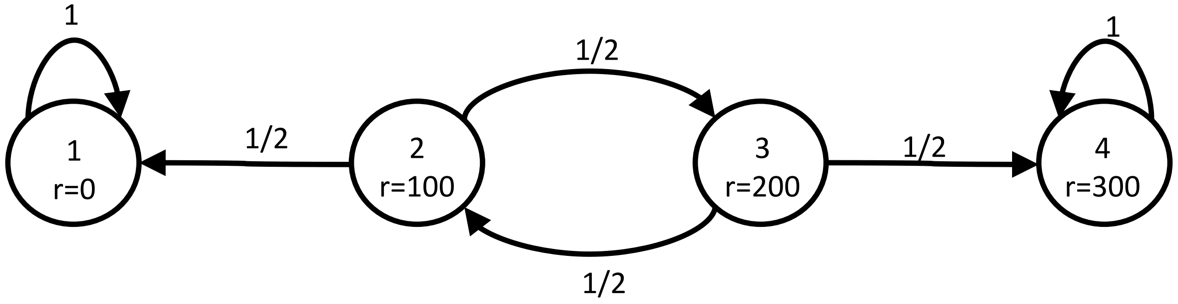
law of total probability

- hence
  - $\pi_j^{(n+1)} = \sum_{i \in S} \pi_i^{(n)} P_{ij}$
  - j^{th} element of $\pi^{(n+1)}$ equals product of row vector $\pi^{(n)}$ with j^{th} column of P
  - $\boldsymbol{\pi}^{(n+1)} = \boldsymbol{\pi}^{(n)} \boldsymbol{P}$

- as a result
  - $\pi^{(1)} = \pi^{(0)} P$
  - $\pi^{(2)} = \pi^{(1)} P = \pi^{(0)} P \cdot P = \pi^{(0)} P^2$
  - $\pi^{(3)} = \pi^{(2)} P = \pi^{(0)} P^2 \cdot P = \pi^{(0)} P^3$
  - $\dots$
  - $\boldsymbol{\pi}^{(n)} = \boldsymbol{\pi}^{(0)} \boldsymbol{P}^n$

example – gambler's ruin

- initial capital €100
  - $\pi^{(0)} = [0, 1, 0, 0]$
- distribution after 10 spins of wheel
  - $\pi^{(10)} = \pi^{(0)} P^{10} \approx [0.667, 0.001, 0.000, 0.333]$
- win probability after 10 spins
  - $\pi_4^{(10)} \approx 0.333$
- expected amount of cash after 10 spins
  - $\pi^{(10)} r^T = \pi^{(10)} [0,100,200,300]^T \approx$ €100



$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P^{10} \approx \begin{bmatrix} 1.000 & 0.000 & 0.000 & 0.000 \\ 0.667 & 0.001 & 0.000 & 0.333 \\ 0.333 & 0.000 & 0.001 & 0.667 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix}$$

transient analysis– exercises

- Section B.3 in the course notes
  - Exercise B.9 (Probability distributions via matrix algebra)
    - use CMWB (DTMC) to answer (c) and (d)
      1. create the model corresponding to the given probability matrix
      2. select ‘Transient Distribution’ button and enter the number of steps to analyze
- answers are provided in Section B.8 of the course notes

interpretation P_{ij}^{n}: n-step transition probabilities

- $P_{ij}^{n} = P(X_{m+n} = j \mid X_m = i)$
  - probability that system transits from state *i* to state *j* in *n* steps
  - P_{ij}^{n} is the **n-step transition probability** from state *i* to state *j*
  - P^{n} is the **n-step probability matrix**
  - $P^{0} = I$ where I denotes the **identify matrix**

interpretation P_{ij}^n: sum of probabilities of paths

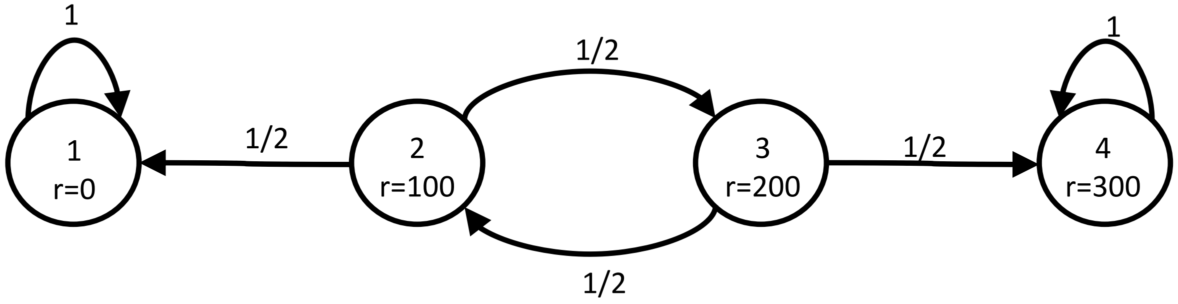
- A **path** is a sequence of states $i_1, i_2, \cdots, i_n$
  - where $n \geq 1$ and $P_{i_m i_{m+1}} > 0$ for all $\mathrm{m} \in \{1, \cdots, n-1\}$
  - called path from i_1 to i_n of length $n-1$ (length equal to number of transitions made)

- $P(i_1, i_2, \cdots, i_n)$ is the probability of path $i_1, i_2, \cdots, i_n$
  - $P(i_1, i_2, \cdots, i_n) = \prod_{m=1}^{n-1} P_{i_m i_{m+1}}$

- $P_{ij}^n = \sum \{ P(i, i_1, \cdots, i_{n-1}, j) \mid i, i_1, \cdots i_{n-1}, j \text{ is a path of length } n \}$
  - sum of probabilities of all paths from i to j of length n

example – gambler's ruin

- probability gambler wins after 10 spins, starting with €200
- By matrix multiplication
  - $P_{34}^{10} \approx 0.667$
- By counting paths
  - path *3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4*: $\frac{1}{2}$
  - path *3, 2, 3, 4, 4, 4, 4, 4, 4, 4, 4*: $\frac{1}{2}^3$
  - path *3, 2, 3, 2, 3, 4, 4, 4, 4, 4, 4*: $\frac{1}{2}^5$
  - path *3, 2, 3, 2, 3, 2, 3, 4, 4, 4, 4*: $\frac{1}{2}^7$
  - path *3, 2, 3, 2, 3, 2, 3, 2, 3, 4, 4*: $\frac{1}{2}^9$
  - $\frac{1}{2} + \frac{1}{2}^3 + \frac{1}{2}^5 + \frac{1}{2}^7 + \frac{1}{2}^9 \approx 0.667$



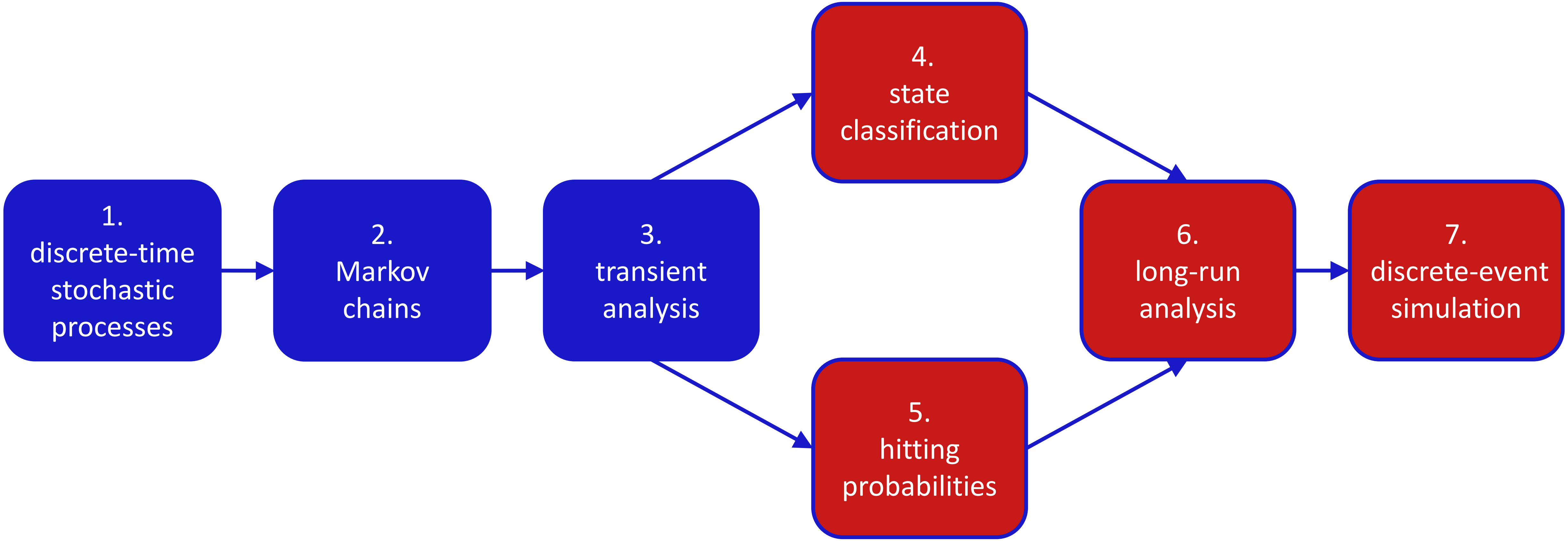
$$\mathrm{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathrm{P}^{10} \approx \begin{bmatrix} 1.000 & 0.000 & 0.000 & 0.000 \\ 0.667 & 0.001 & 0.000 & 0.333 \\ 0.333 & 0.000 & 0.001 & 0.667 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix}$$

transient analysis – exercises

- Section B.3 in the course notes
  - Exercise B.10 (N-step transition probabilities)
    - use CMWB (DTMC)
      1. create the model corresponding to the given probability matrix
      2. select ‘Transient Matrix’ and enter the number of steps (n) to compute the n-step transition matrix
  - Exercise B.11 (Gambler’s ruin - n-step probabilities and expected reward)
    - use CMWB (DTMC) to answer (a) and (c); for (c)
      1. create a copy of the Gambler’s ruin model and adapt the initial distribution
      2. selected ‘Transient Rewards’ and enter the number of steps to compute the expected reward
  - Exercise B.12 (Gambler’s ruin – dependent and non-identically distributed variables)
    - use CMWB (DTMC)
  - Exercise B.13 (Queue in time-slotted communication network – modeling and transient analysis)
    - use CMWB (DTMC)
  - Exercise B.14 (Independent identically distributed variables as Markov chain)
- answers are provided in Section B.8 of the course notes

module B - submodules and dependencies



lessons learned

- distribution vectors can be computed by
  - $\pi^{(n+1)} = \pi^{(n)} P$
  - $\pi^{(n)} = \pi^{(0)} P^n$
- if distribution vector is known, expected reward is know as well
  - $\mathrm{E}\big(\mathrm{r}(\mathrm{X_n})\big) = \pi^{(n)} \cdot r^T$
- **n-step transition probability** from state i to state j equals P_{ij}^n
  - P^n is the **n-step transition probability matrix**
  - P_{ij}^n equals sum of probabilities of all **paths** from i to j of length n

ES ELECTRONIC SYSTEMS TU/e

questions

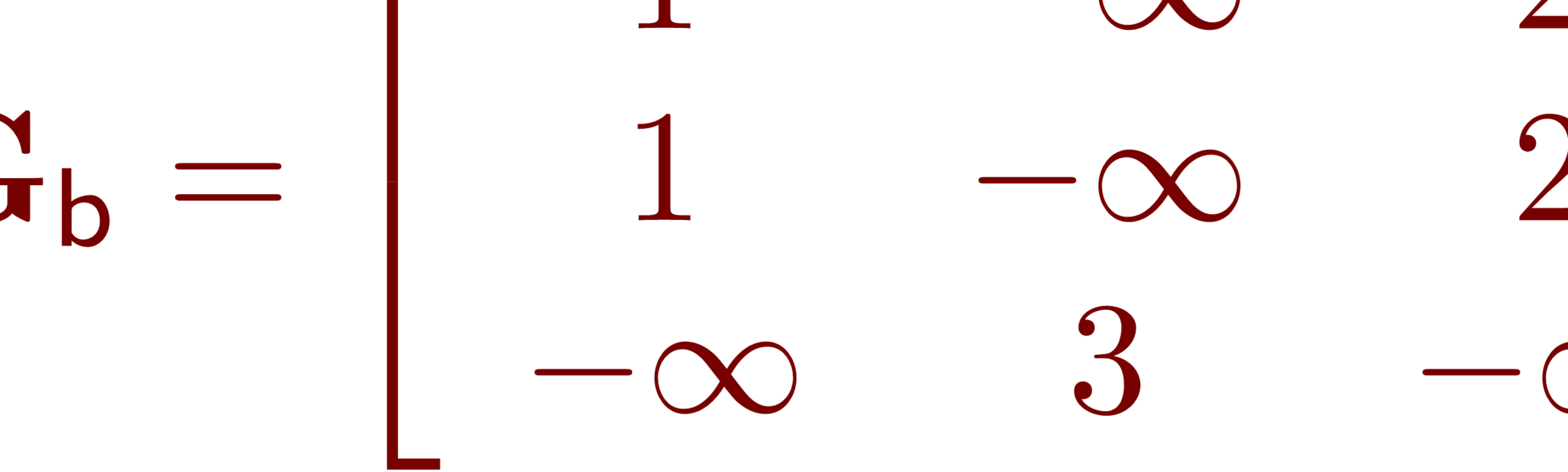
- can we always compute probability distributions and expected rewards? no, but we can do so for sequences of independent identically distributed variables and for Markov chains
- how to compute the probability vector (and expected reward) at time n?
- how to compute the probability that the Markov chain transits from state i to state j in n steps?

questions

- can we always compute probability distributions and expected rewards? no, but we can do so for sequences of independent identically distributed variables and for Markov chains
- how to compute the probability vector (and expected reward) at time n? by multiplying the initial distribution with the n^{th} power of the transition probability matrix; the expected reward is the product of this vector with the reward vector
- how to compute the probability that the Markov chain transits from state i to state j in n steps? by computing the n^{th} power of the transition probability matrix and taking the element on row i and column j or by summing the probabilities of all paths from i to j

questions

- can we also compute the limiting behavior when n grows very large? yes, but
  i. the limiting behavior depends on the structural properties of the chain, and
  ii. requires the computation of hitting probabilities
- what are the structural properties of Markov chains on which the limiting behavior depends?
- what are hitting probabilities and how to compute them?



B.4 – state classification

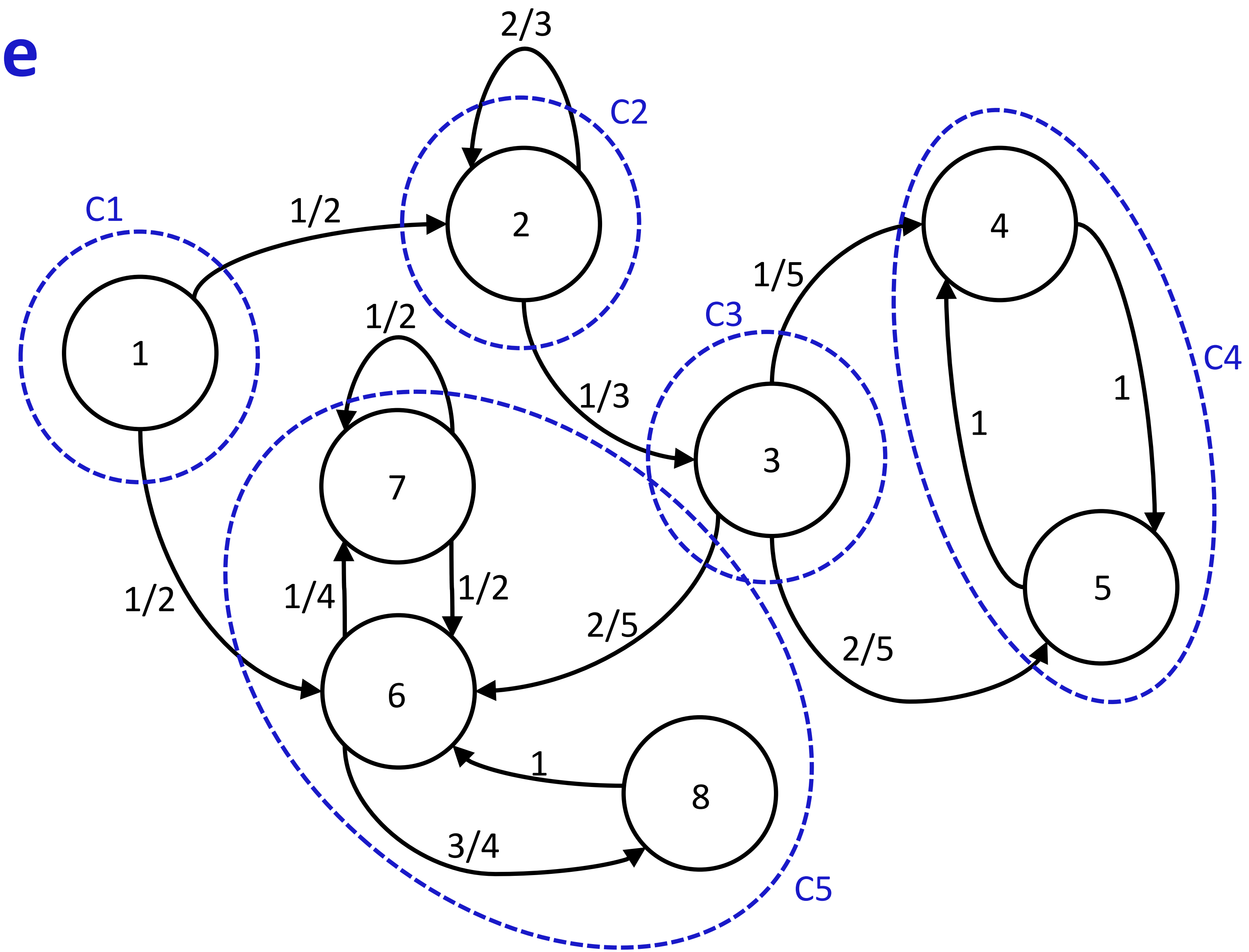
equivalence relation on states

- long-run / limiting behavior of Markov chains depends on the structure of the transition matrix
  - capture structure as equivalence relation

- state j is **accessible** from state i, written $i \to j$, if and only if $P_{ij}^n > 0$ for some $n \geq 0$
  - so $i \to j$ if and only a path exists from i to j
  - note that $i \to i$

- states i and j **communicate**, written $i \leftrightarrow j$, if and only $i \to j$ and $j \to i$
  - relation $\leftrightarrow$ is an **equivalence relation** (reflexive, symmetric and transitive)
  - partitions state-space into **classes** of states (disjoint, non-empty)

example



state classification – exercises

- Section B.4 in the course notes
  - Exercise B.15 (Classes of a Markov chain)
    - use CMWB (DTMC) to verify your answer
      1. create the model corresponding to the given probability matrix
      2. use ‘View Transition Diagram’ transition diagram to identify classes
      3. select ‘Communication Classes’ to compute the classes
  - Exercise B.16 (State accessibility versus paths)
  - Exercise B.17 (Communicating states – equivalence relation)
- answers are provided in Section B.8 of the course notes

recurrent and transient states

- a state i is **recurrent** if and only if $f_{ii} = 1$
  - $f_{ii} = \sum \{ P(i, i_1, \cdots, i_{n-1}, i) \mid i, i_1, \cdots i_{n-1}, i \text{ is a path of length } n \geq 1 \text{ s.t. } i_k \neq i \ (k = 1, \cdots, n-1)\}$
  - f_{ii} is the *return probability* that the chain, starting from i will eventually visit i again in one or more steps
  - when a chain visits a recurrent state, it will keep returning to it infinitely often

- a state i is **transient** if and only if $f_{ii} < 1$
  - a transient state is only visited a finite number of times; at some point in time it will not be visited anymore

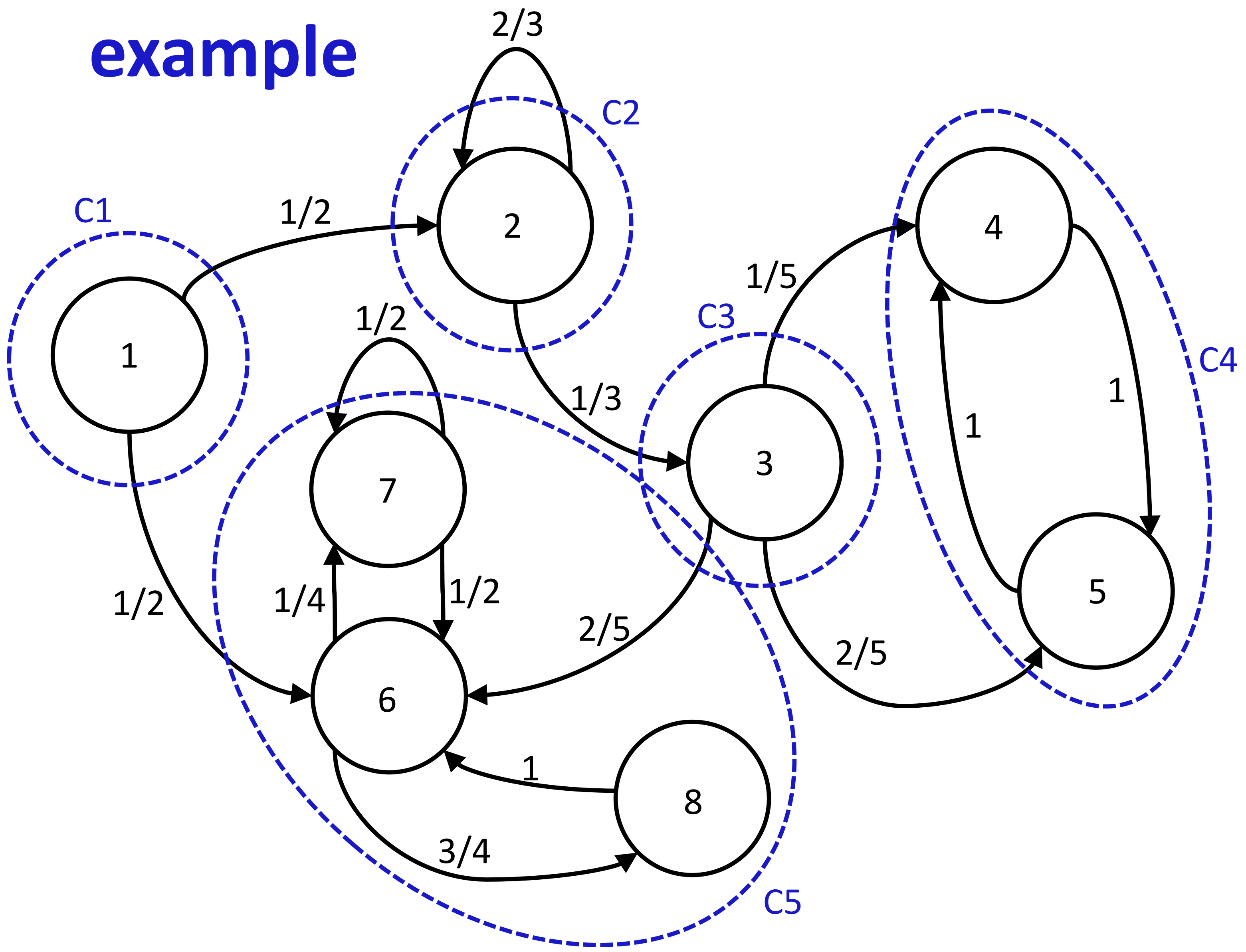
recurrent and transient classes

- recurrence and transience are **class properties**
  - the states in a class are either all recurrent or all transient
  - so we will refer to recurrent or transient classes of states

- a class without outgoing transitions leading to another class is called **closed**

- a class is recurrent if and only if it is closed
  - this property holds for finite-state Markov chains

example



C1:

- **transient**
- $f_{11} = 0$

C2:

- **transient**
- $f_{22} = \frac{2}{3}$

C3:

- **transient**
- $f_{33} = 0$

C4:

- **recurrent**
- $f_{44} = 1$

C5:

- **recurrent**
- $f_{77} = \frac{1}{2} + \sum_{i=0}^{\infty} \frac{1}{2} \cdot \left(\frac{3}{4}\right)^{i} \cdot \frac{1}{4} = 1$

state classification – exercises

- Section B.4 in the course notes
  - Exercise B.18 (Recurrent versus transient classes)
    - use CMWB (DTMC) to check answer
      1. create the model corresponding to the given probability matrix (same as Exercise B.15)
      2. select ‘Classify Transient Recurrent’ to determine whether states are transient or recurrent
  - Exercise B.19 (Computing return probabilities through paths)
    - possibly use CMWB (DTMC) to create transition graph (same as Exercise B.15)
- answers are provided in Section B.8 of the course notes

periodic and aperiodic states

- the **period** of a *recurrent* state i, written $d(i)$, is defined as
  - $d(i) = gcd\{ n \geq 1 \mid P_{ii}^{n} > 0\}$
  - $d(i) = gcd\{ n \geq 1 \mid \text{a path of length } n \text{ exists from } i \text{ to } i\}$

- hence when state i is visited than
  - the number of steps for which return to i is possible is a multiple of $d(i)$
  - returning to i in *m* steps is not possible if *m* is no multiple of $d(i)$

periodic and aperiodic states

- when $d(i)$ >1 certain revisiting times are excluded
  - if *i* is visited at time *n* it can e.g. not be visited at time *n+1* since *1* is not a multiple of *d(i)*
  - such states are called **periodic** since they are visited in period patterns

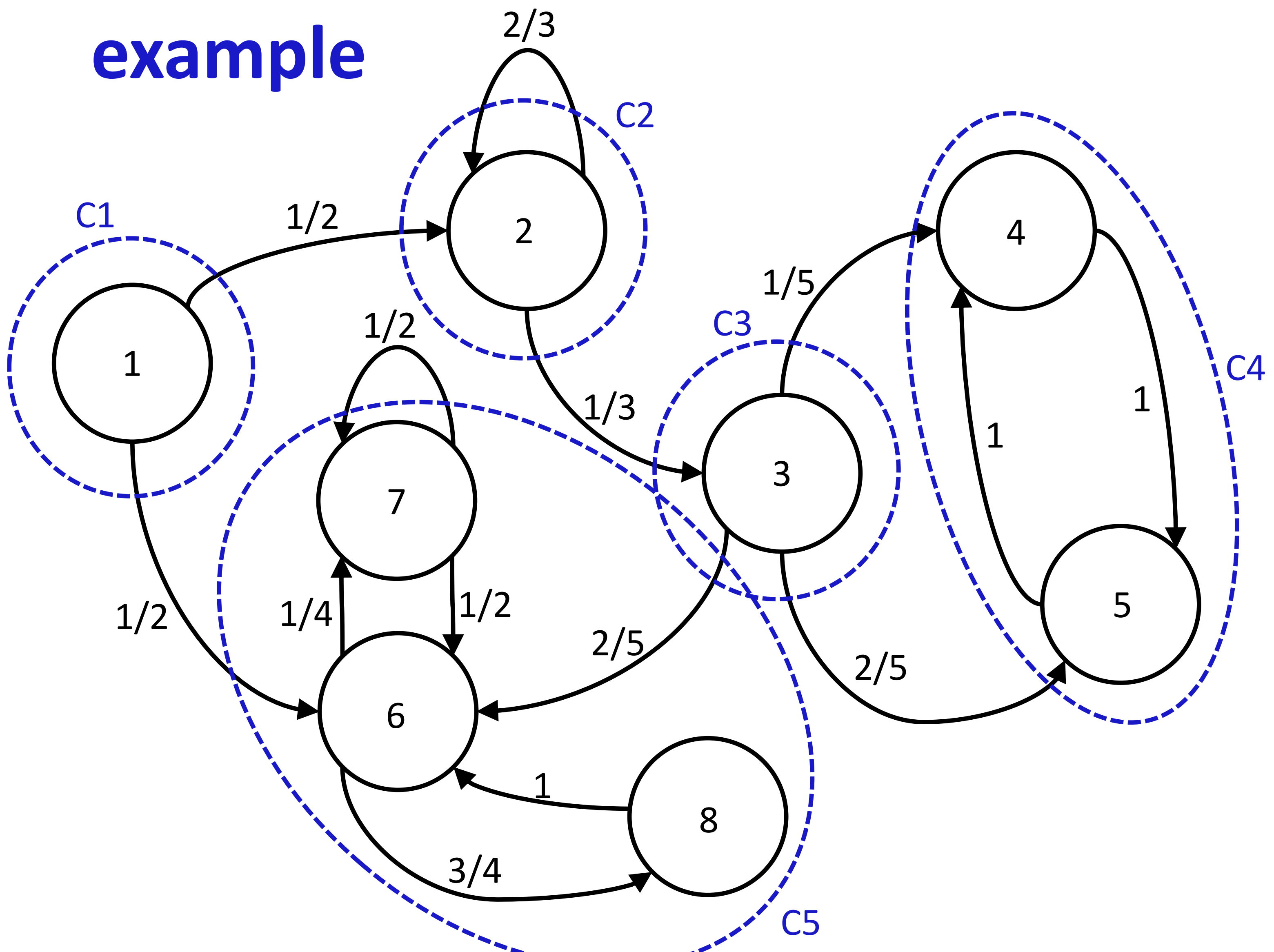
- when $d(i)$ =1 no revisiting times are excluded
  - if *i* is visited at time *n*, from time $n + (N-1)^2 - 1$ onwards it has a positive probability of being revisited
  - such states are called **aperiodic**

periodic and aperiodic classes

- periodicity is a class property
  - all states in a recurrent class have the same period
  - states in a recurrent class are either all aperiodic or all periodic
  - lift the concept of periodicity to the level of classes

- for any aperiodic class C and all states $\mathrm{i}, \mathrm{j} \in C$
  - $P_{ij}^{n} > 0$ for all $n \geq (N-1)^2 + 1$
  - some time after class entrance, the chain can be visiting any state in the class with positive probability
  - memory about entry time and state appear to fade away

example



C1:
- **transient**

C2:
- **transient**

C3:
- **transient**

C4:
- **recurrent**
- **periodic (period 2)**

C5:
- **recurrent**
- **aperiodic (state 7 has period 1)**

terminology

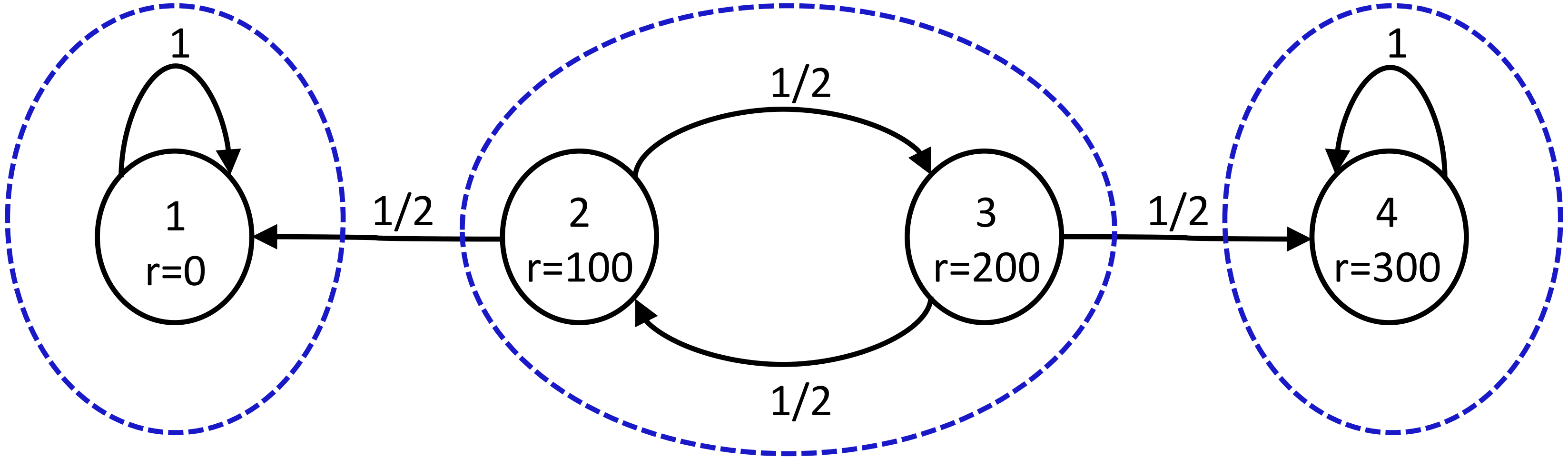
- a *class* is **ergodic** if it is both recurrent and aperiodic
  - a class that is not ergodic is called **non-ergodic**
- a *Markov chain* is **ergodic** if all its recurrent classes are ergodic
  - a *Markov chain* that is not ergodic is called **non-ergodic**

- a **unichain** is a *Markov chain* with a single recurrent class (and zero or more transient ones)
  - a unichain visits transient states a finite number of times and eventually ends up in its recurrent class
  - a *Markov chain* that is not a unchain is called a **non-unichain**

Terminology: Markov chain types

Markov chain type	unichain	non-unichain
ergodic	a single recurrent class which is aperiodic (and zero or more transient classes)	at least two recurrent classes, all of which are aperiodic (and zero or more transient classes)
non-ergodic	a single recurrent class which is periodic (and zero or more transient classes)	at least two recurrent classes, which are not all aperiodic (and zero or more transient classes)

example: gambler's ruin



- **recurrent**
- **aperiodic**
- **ergodic**

- **transient**

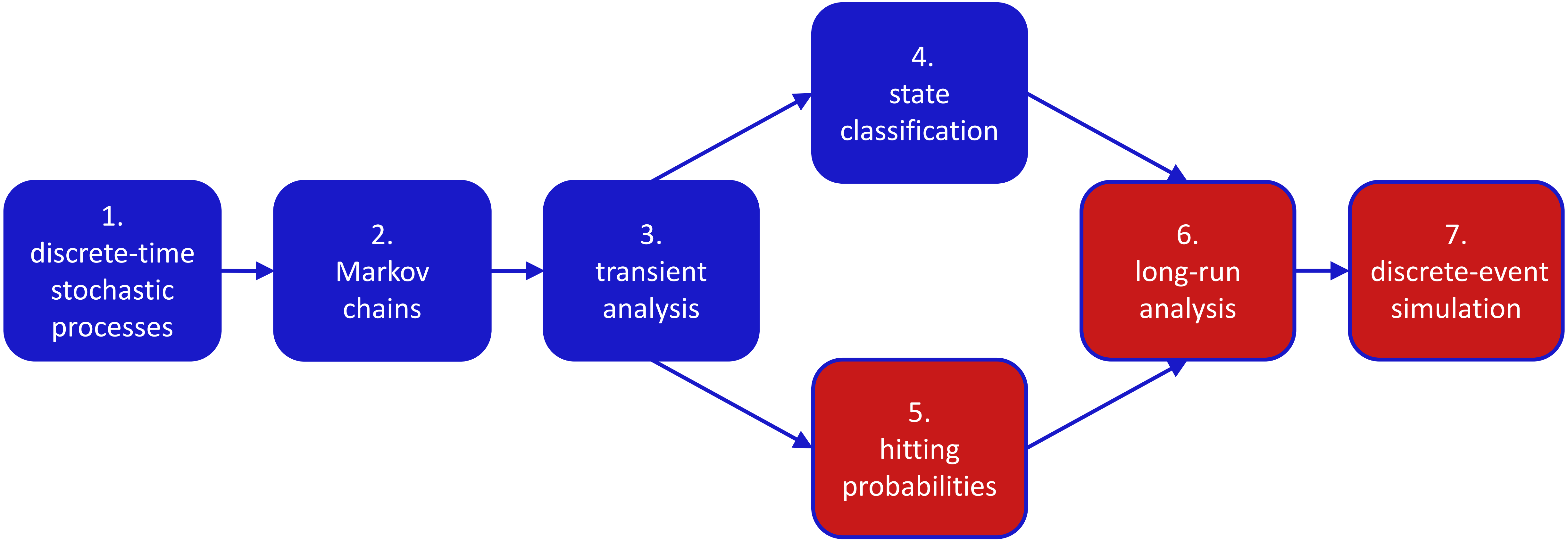
- **recurrent**
- **aperiodic**
- **ergodic**

- gambler's ruin
  - ergodic non-unichain since it contains two recurrent aperiodic classes
  - in the long run the chain will jump to one of the two recurrent classes and remain there forever

state classification – exercises

- Section B.4 in the course notes
  - Exercise B.20 (Periodic versus aperiodic classes)
    - use CMWB (DTMC) to check answer
      1. create the model corresponding to the given probability matrix
      2. select 'Determine Periodicity' to determine whether states are aperiodic or periodic, and in the later case compute the period
      3. select 'Determine MC Type' to obtain information about the Markov chain type
  - Exercise B.21 (Aperiodic states are eventually visited)
    - use CMWB (DTMC) to aid in solving (b)
- answers are provided in Section B.8 of the course notes

module B - submodules and dependencies



lesson learned

- the state-space of a Markov chain is partitioned in **classes** of **communicating states**
- states and classes are either **recurrent** or **transient**
  - states in a class are either all recurrent or all transient
- recurrent states and classes are either **periodic** or **aperiodic**
  - states in a class are either all periodic or all aperiodic
- a Markov chain is either a **unichain** or a **non-unichain**
- a Markov chain is either a **ergodic** or **non-ergodic**

questions

- can we compute the limiting behavior when n grows very large? yes, but
  i. the limiting behavior depends on the structural properties of the chain, and
  ii. requires the computation of hitting probabilities
- what are the structural properties of Markov chains on which the limiting behavior depends?
- what are hitting probabilities and how to compute them?

questions

- can we compute the limiting behavior when n grows very large? yes, but
  i. the limiting behavior depends on the structural properties of the chain, and
  ii. requires the computation of hitting probabilities
- what are the structural properties of Markov chains on which the limiting behavior depends? the limiting behavior depends on whether the chain is ergodic or non-ergodic and whether the chain is a unichain or a non-unichain
- what are hitting probabilities and how to compute them?

$$G_b = \begin{bmatrix} 1 & \infty & 2 \\ 1 & -\infty & 2 \\ -\infty & 3 & -\infty \end{bmatrix}$$

B.5 – hitting probabilities

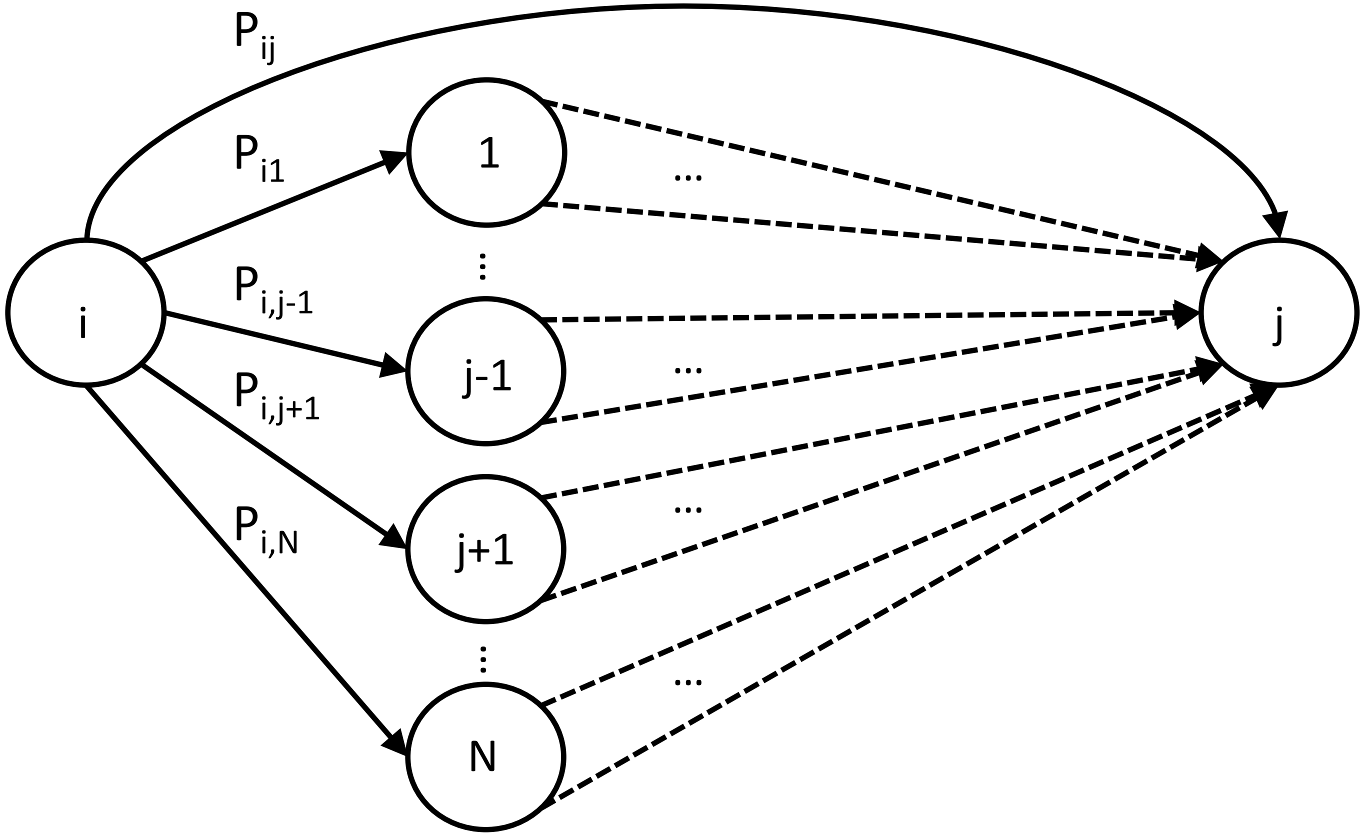
probability to hit a state

- $f_{ij} = \sum \{ P(i, i_1, \cdots, i_{n-1}, j) \mid i, i_1, \cdots i_{n-1}, j \text{ is a path of length } n \geq 1 \text{ s.t. } i_k \neq i \ (k = 1, \cdots, n-1)\}$
  - **hit probability** that **state** j is ever visited, starting from state i in *one or more* steps
  - return probability f_{ii} is a special case of hit probability

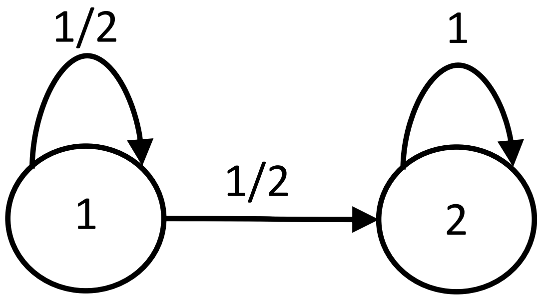
- in general hit probabilities are hard to compute
  - establish more convenient way based on solving system of linear equations

solution to system of linear equations

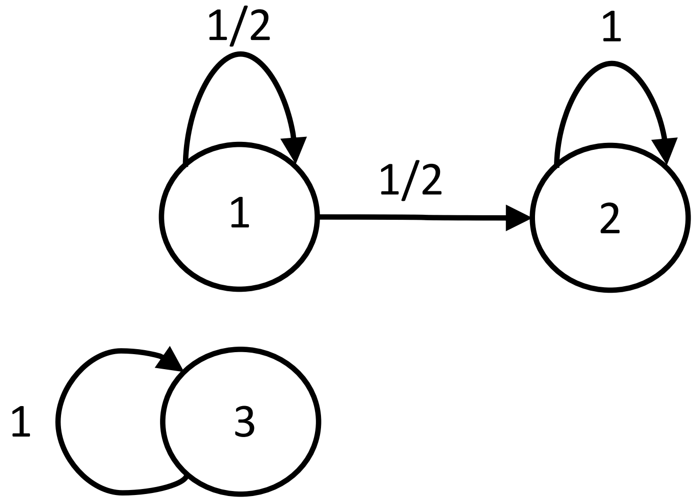
- decompose set of paths (see picture)
- so $f_{ij} = P_{ij} + \sum_{k \in S \setminus \{j\}} P_{ik} f_{kj}$
- fix j and replace each f_{ij} by variable x_i
  - for each $i \in S$
- result: system of linear equations
  - $x_i = P_{ij} + \sum_{k \in S \setminus \{j\}} P_{ik} x_k$
- f_{ij} form a solution to system of linear equations



example: hit state 2



- via paths: $f_{12} = \sum_{i=1}^{\infty} \frac{1}{2}^{i} = 1, f_{22} = 1$
- equations: $x_1 = \frac{1}{2} + \frac{1}{2}\mathrm{x}_1, x_2 = 1$
- solution: $x_1 = 1, x_2 = 1$



- via paths: $f_{12} = \sum_{i=1}^{\infty} \frac{1}{2}^{i} = 1, f_{22} = 1, f_{32} = 0$
- equations: $x_1 = \frac{1}{2} + \frac{1}{2}\mathrm{x}_1, x_2 = 1, x_3 = x_3$
- solution: $x_1 = 1, x_2 = 1, x_3 \in \mathcal{R}$

least non-negative solution to linear equations

- system of equation can have multiple solutions
- but f_{ij} form the *least non-negative* solution
- assume $y_i \geq 0$ forms a solution, then $y_i = P_{ij} + \sum_{k \in S \backslash \{j\}} P_{ik} y_k$
- We have to show that $y_i \geq f_{ij}$
- repeated substitution yields
  - $y_i \geq P_{ij}$ (probability of path of length 1)

    $+ \sum_{k_1 \in S \backslash \{j\}} P_{ik_1} P_{k_1 j}$ (probability of paths of length 2)

    $+ \sum_{k_1 \in S \backslash \{j\}} \sum_{k_2 \in S \backslash \{j\}} P_{ik_1} P_{k_1 k_2} P_{k_2 j}$ (probability of paths of length 3)

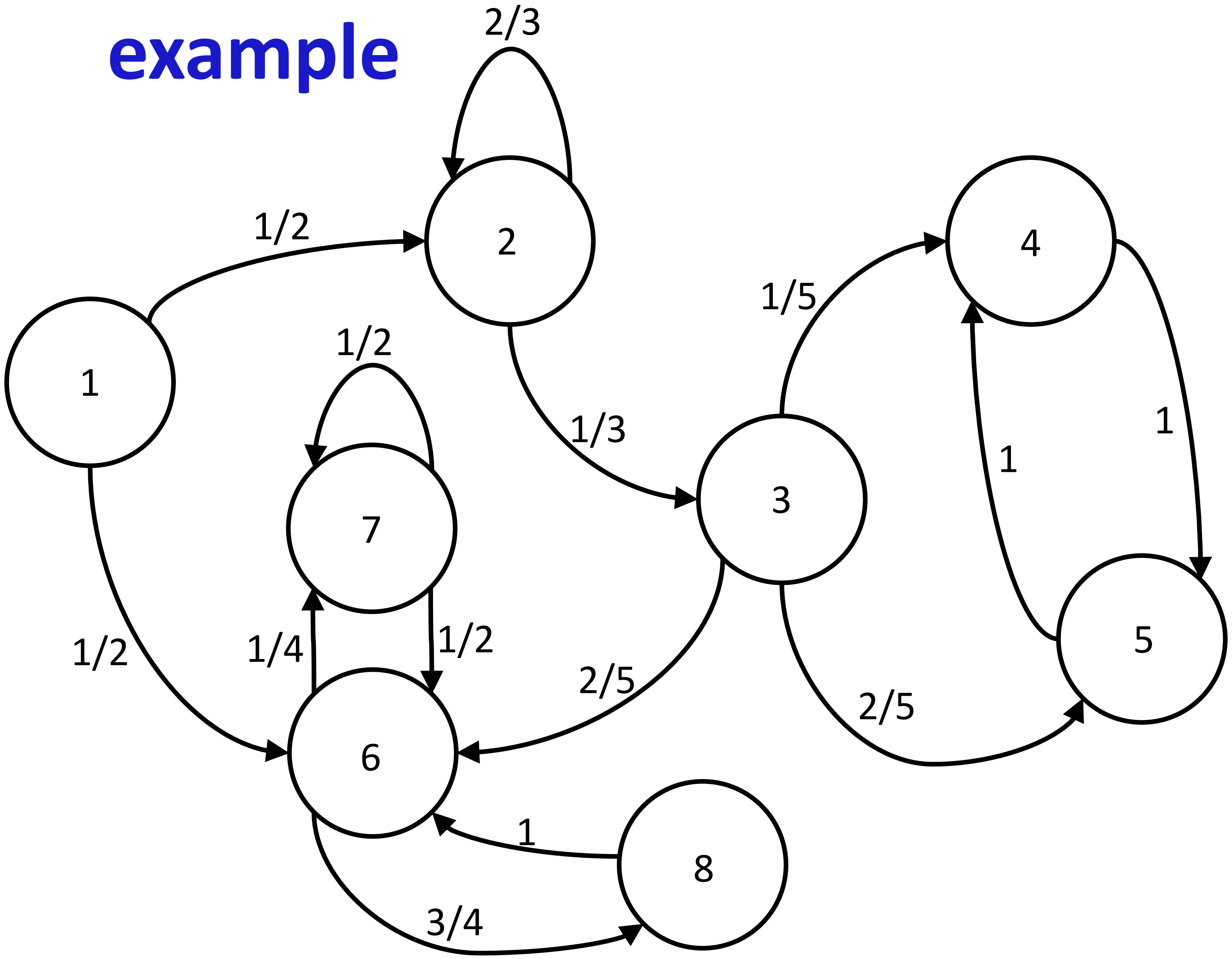
    $+ \cdots$

    $+ \sum_{k_1 \in S \backslash \{j\}} \sum_{k_2 \in S \backslash \{j\}} \cdots \sum_{k_n \in S \backslash \{j\}} P_{ik_1} P_{k_1 k_2} \cdots P_{k_n j}$ (probability of paths of length n)

    $\cancel{+ \sum_{k_1 \in S \backslash \{j\}} \sum_{k_2 \in S \backslash \{j\}} \cdots \sum_{k_n \in S \backslash \{j\}} \sum_{k_{n+1} \in S \backslash \{j\}} P_{ik_1} P_{k_1 k_2} \cdots P_{k_n k_{n+1}} y_{k_{n+1}}}$ ~~(residual)~~

- for $n \to \infty$ the sum of probabilities converges to f_{ij} and hence $y_i \geq f_{ij}$

example



- probability to hit state 6 from 1
  - $x_1 = \frac{1}{2}\, x_2 + \frac{1}{2}$
  - $x_2 = \frac{2}{3}\, x_2 + \frac{1}{3} x_3$
  - $x_3 = \frac{1}{5}\, x_4 + \frac{2}{5} x_5 + \frac{2}{5}$
  - $x_4 = x_5,\ x_5 = x_4$
  - $x_6 = \frac{1}{4}\, x_7 + \frac{3}{4} x_8,\ x_7 = \frac{1}{2}\, x_7 + \frac{1}{2}, x_8 = 1$
- least non-negative solution
  - if state 6 is not accessible from i: set $x_i = 0$
  - $x_4 = x_5 = 0$
  - $x_3 = \frac{2}{5}, x_2 = \frac{2}{5}$
  - $x_6 = x_7 = x_8 = 1$
  - $x_1 = \frac{7}{10}$, so $f_{16} = \frac{7}{10}$

hitting probabilities – exercises

- Section B.5 in the course notes
  - verify Example B.11 (Hitting probabilities – hitting a state) (same as example previous slide)
    - use CMBW (DTMC) to check answer
      1. create the model (same model as Example B.7)
      2. select ‘Hitting Probability’ and enter state to hit
  - Exercise B.22 (Computing return probabilities through equations)
    - use CMBW (DTMC) to check answer
      1. create the model corresponding to the given probability matrix (same model as Exercise B.15)
      2. select ‘Hitting Probability’ and enter state to hit
  - Exercise B.23 (Infinite closed classes are not necessarily recurrent)
- answers are provided in Section B.8 of the course notes

expected cumulative reward until hit - definition

- $f_{ij}^r = \sum \{\, P(i, i_1, \cdots, i_{n-1}, j) \cdot (r(i) + r(i_1) + \cdots + r(i_{n-1})) \mid$
  $i, i_1, \cdots i_{n-1}, j \text{ is a path of length } n \geq 1 \text{ s.t. } i_k \neq i \ (k = 1, \cdots, n-1) \,\}$
  - probability-weighted average of cumulative rewards of paths (of length ≥ 1) from i to j

- $\frac{f_{ij}^r}{f_{ij}}$ is **expected cumulative reward** until **state** j is hit, starting from state i
  - is normalized to hit probability to take into account the fact that j is not necessarily hit
  - so expected cumulative reward is conditional on assumption that that j is hit

- interpretation: if we would simulate a large number of paths from i to j, the average of their cumulative rewards would be close to $\frac{f_{ij}^r}{f_{ij}}$

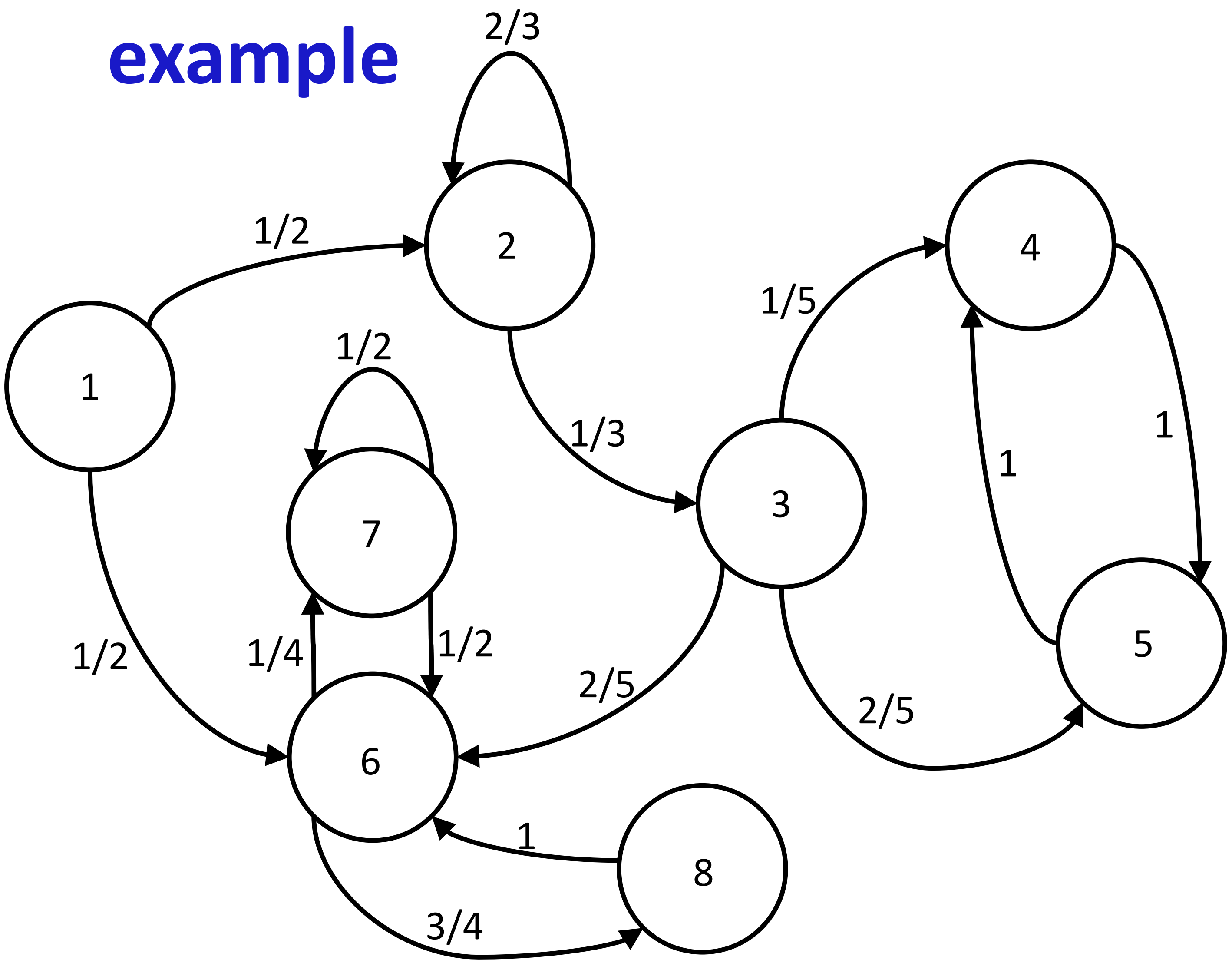
expected cumulative reward until hit - computation

- how to compute the expected cumulative reward?

- probability-weighted average of cumulative rewards f_{ij}^{r} form the least non-negative solution to system of linear equation
  - $x_i = r(i) \cdot f_{ij} + \sum_{k \in S \setminus \{j\}} P_{ik} x_k$ for fixed j and variables x_i for each $i \in S$

- normalize to hitting probability f_{ij}
  - computed by solving $x_i = P_{ij} + \sum_{k \in S \setminus \{j\}} P_{ik} x_k$

example



- expected number of steps to hit state 6, starting from 1
- assign reward 1 to each state
- compute f_{16}^r by solving
  - $x_1 = 1 \cdot f_{16} + \frac{1}{2}\, x_2$
  - $x_2 = 1 \cdot f_{26} + \frac{2}{3}\, x_2 + \frac{1}{3} x_3$
  - $x_3 = 1 \cdot f_{36} + \frac{1}{5}\, x_4 + \frac{2}{5} x_5$
  - $x_4 = x_5,\ \ x_5 = x_4$
- solution
  - $x_1 = \frac{15}{10}$, so $f_{16}^r = \frac{15}{10}$
  - the expected number of steps to hit state 6 from state 1 is thus $\frac{f_{16}^r}{f_{16}} = \frac{\frac{15}{10}}{\frac{7}{10}} = \frac{15}{7}$

hitting probabilities– exercises

- Section B.5 in the course notes
  - verify Example B.12 (Expected cumulative reward – hitting a state) (same as example previous slide)
    - use CMWB (DTMC) to check answers
      1. create the model (same model as Example B.7) and add rewards
      2. select 'Reward Until Hit' and enter state to hit
  - Exercise B.24 (Hiccups in video application)
    - use CMWB (DTMC) to check answers
      1. create the Markov reward model for different values of parameter p
      2. select 'Reward Until Hit' and enter state to hit
- answers are provided in Section B.8 of the course notes

probability to hit a set of states

- $h_{iH} = \sum \{ P(i, i_1, \cdots, i_n) \mid i, i_1, \cdots i_n$ is a path of length $n \geq 0$ s.t.

  $i \in H$ implies $n = 0$ and

  $i \notin H$ implies $n \geq 1$, $i_n \in H$ and $i_k \notin H$ for all $\mathrm{k} = 1 \cdots n - 1 \}$

  - **hit probability** that any of the states in **state set** $H \subseteq S$ is ever visited, starting from state i in *zero or more* steps

- probabilities h_{iH} form the least non-negative solution to system of equations

  - $x_i = \begin{cases} 1 & \text{if } i \in H \\ \sum_{k \in S} P_{ik} x_k & \text{if } i \notin H \end{cases}$

expected cumulative reward to hit a set of states

- $h_{iH}^r = \sum \{ P(i, i_1, \cdots, i_n) \cdot (r(i) + r(i_1) + \cdots + r(i_{n-1})) | \, i, i_1, \cdots i_n$ is a path of length $n \geq 1$ s.t. $i \notin H, i_n \in H$ and $i_k \notin H$ for all $\mathrm{k} = 1 \cdots n-1 \}$
  - probability-weighted average of cumulative rewards of paths (of length ≥ 0) from i to H

- $\frac{h_{iH}^r}{h_{iH}}$ is **expected cumulative reward** until **state set** H is hit, starting from state i

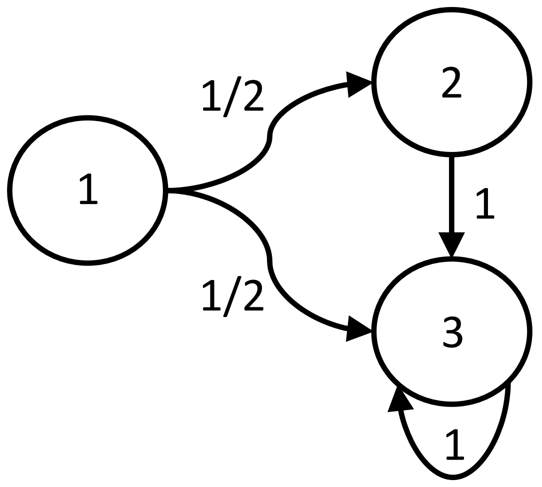
- probability-weighted average of cumulative rewards h_{iH}^r form the least non-negative solution to system of linear equation
  - $x_i = \begin{cases} 0 & i \in H \\ r(i) \cdot h_{iH} + \sum_{k \in S} P_{ik} x_k & i \notin H \end{cases}$

hitting a state versus hitting a set of states

- f_{ii} is in general different from $h_{i\{i\}}$
  - $h_{i\{i\}} = 1$
  - in case i is a recurrent state, $f_{ii} = 1$
  - in case i is a transient state, $f_{ii} < 1$

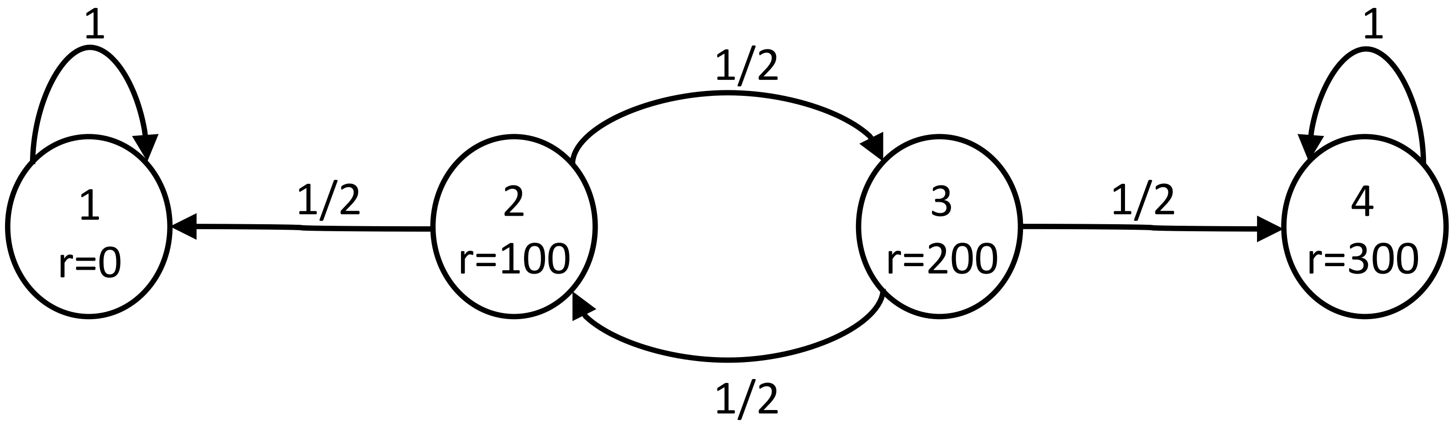
- $f_{ij} = h_{i\{j\}}$ if $i \neq j$

- h_{iH} is in general different from $\sum_{j \in H} f_{ij}$
  - $\sum_{j \in H} f_{ij}$ can count prefixes of paths multiple times



- $h_{1\{2,3\}} = \frac{1}{2} + \frac{1}{2} = 1$
  - paths 1, 2 and 1, 3
- $f_{12} = \frac{1}{2}$
  - path 1, 2
- $f_{13} = \frac{1}{2} + \frac{1}{2} = 1$
  - path 1, 2, 3 and 1, 3

example: gambler's ruin

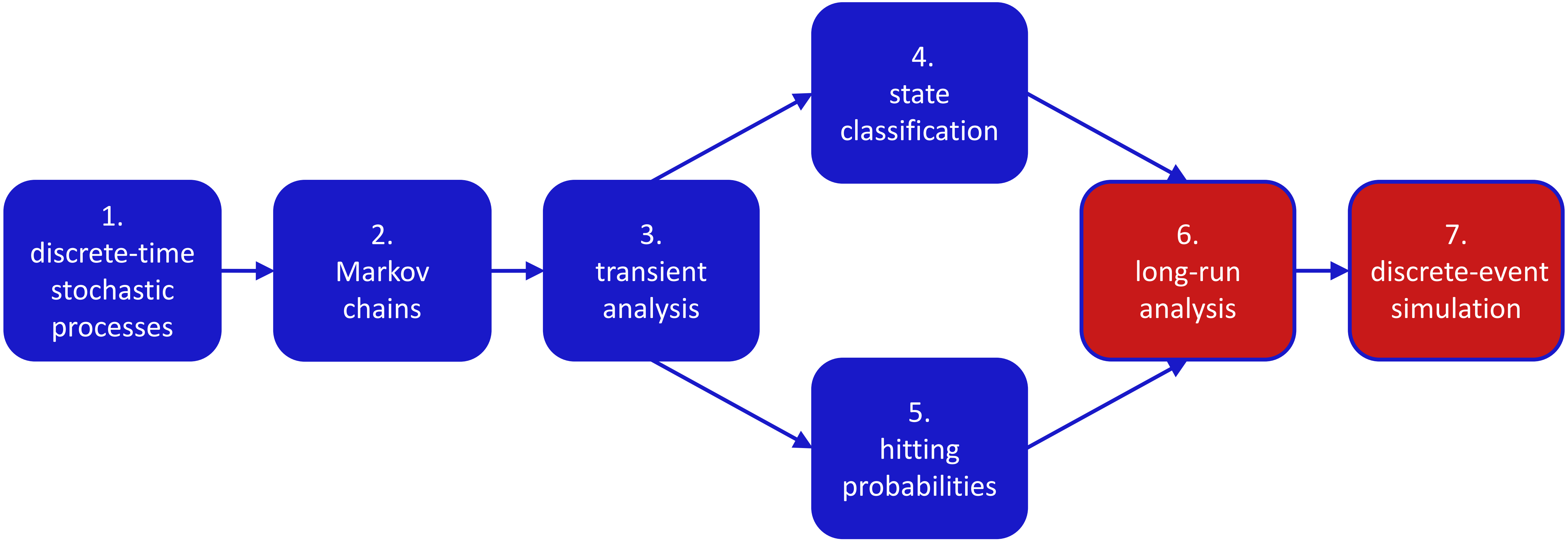


- expected number of spins required before gambler wins, starting with initial capital of €100
  - define new reward r' assigning value 1 to each state
  - define hit set $H = \{4\}$
  - compute $\frac{h^{r'}_{2H}}{h_{2H}}$
- equations for $h^{r'}_{2H}$
  - $x_1 = 1 \cdot h_{1H} + x_1$ $\quad (h_{1H} = 0)$ $\quad (x_1 = 0)$
  - $x_2 = 1 \cdot h_{2H} + \frac{1}{2} x_1 + \frac{1}{2} x_3$ $\quad (h_{2H} = \frac{1}{3})$ $\quad (x_2 = \frac{8}{9})$
  - $x_3 = 1 \cdot h_{3H} + \frac{1}{2} x_2 + \frac{1}{2} x_4$ $\quad (h_{3H} = \frac{2}{3})$ $\quad (x_3 = \frac{10}{9})$
  - $x_4 = 0$
- solution: $\frac{8}{9} / \frac{1}{3} = 2\,\frac{2}{3}$

hitting probabilities – exercises

- Section B.5 in the course notes
  - verify Example B.13 (Hitting probabilities – hitting a set)
    - use CMWB (DTMC) to check answer
      1. create the model (same model as Example B.7)
      2. select ‘Hitting Probability Set’ and enter states to hit
  - verify Example B.14 (Gambler’s ruin – win probability) and Example B.15 (Gambler’s ruin - expected number of spins until win)
    - use CMWB (DTMC) to check answer
      1. use Gambler’s ruin model and adapt rewards
      2. select ‘Hitting Probability Set’ / ‘Reward Until Hit Set’ and enter states to hit
  - Exercise B.25 (Hitting a state versus hitting a singleton state set)
  - Exercise B.26 (Hitting recurrent states with probability 1)
  - Exercise B.27 (Rover in a maze)
    - use CMWB (DTMC) to check the answer
- answers are provided in Section B.8 of the course notes

module B - submodules and dependencies



lessons learned

- **hit probability of state j** from state i is the cumulative probability of all paths (with length ≥ 1) from i to j (not visiting i in between)
  - computed by solving system of linear equations
  - special case: return probability
- **hit probability of state set H** from state i is the cumulative probability of all paths (with length ≥ 0) from i to H (not visiting i in between)
  - computed by solving system of linear equations
- **expected cumulative reward until hit** is probability-weighted average of cumulative rewards of paths normalized to hit probability
  - computed by solving systems of linear equations and taking quotient
  - special case: expected return time

questions

- can we compute the limiting behavior when n grows very large? yes, but
  i. the limiting behavior depends on the structural properties of the chain, and
  ii. requires the computation of hitting probabilities
- what are the structural properties of Markov chains on which the limiting behavior depends? the limiting behavior depends on whether the chain is ergodic or non-ergodic and whether the chain is a unichain or a non-unichain
- what are hitting probabilities and how to compute them?

questions

- can we compute the limiting behavior when n grows very large? yes, but
  i. the limiting behavior depends on the structural properties of the chain, and
  ii. requires the computation of hitting probabilities
- what are the structural properties of Markov chains on which the limiting behavior depends? the limiting behavior depends on whether the chain is ergodic or non-ergodic and whether the chain is a unichain or a non-unichain
- what are hitting probabilities and how to compute them? hitting probabilities are the probabilities to eventually reach a state or a set of states from a starting state; hitting probabilities are computed by a solving system of linear equations

questions

- how to compute limiting matrices?
- how to compute limiting distributions?
- how to compute limiting rewards?

$$G_b = \begin{bmatrix} 1 & -\infty & 2 \\ -\infty & 3 & -\infty \end{bmatrix}$$

B.6 – long-run analysis

long-run analysis

- transient analysis: compute *probability distribution* and *expected reward* at time n
  - $\pi^{(n+1)} = \pi^{(n)} \cdot P$
  - $\pi^{(n)} = \pi^{(0)} \cdot P^n$
  - $E\big(r(x_n)\big) = \pi^{(n)} \cdot r^T$

- long-run analysis: what happens if $n \to \infty$
  - probability distribution
  - probability matrix
  - expected reward

ergodic unichain - limiting distribution / expected reward

- limiting distribution $\lim_{n\to\infty} \pi^{(n)} = \pi^{(\infty)}$ exists for every ergodic unichain
  - chain with a single recurrent class which is aperiodic
  - so $X_0, X_1, \cdots$ converges to a random variable with distribution $\pi^{(\infty)}$
  - this mode of converge is called *converge in distribution*

- long-run expected reward exists: $\lim_{n\to\infty} E\big(r(X_n)\big) = \lim_{n\to\infty} \pi^{(n)} \cdot r^T = \pi^{(\infty)} \cdot r^T$

ergodic unichain – limiting distribution / expected reward

- how to compute $\pi^{(\infty)}$?
  - $\pi^{(n+1)} = \pi^{(n)} \cdot P$ and hence $\lim_{n\to\infty} \pi^{(n+1)} = \lim_{n\to\infty} \pi^{(n)} \cdot P$ and thus $\pi^{(\infty)} = \pi^{(\infty)} \cdot P$
  - hence $\pi^{(\infty)}$ is a solution to the so-called **balance equations**: $\pi = \pi \cdot P, \sum_{i\in S} \pi_i = 1$

- balance equations of an ergodic unichain have a unique solution
  - a solution to the balance equations is called a **stationary solution**

ergodic unichain - limiting matrix

- we know $\pi^{(n)} = \pi^{(0)} \cdot P^n$ and thus $\pi^{(\infty)} = \pi^{(0)} \cdot P^\infty$
  - where $P^\infty = \lim_{n \to \infty} P^n$

- hence $\pi^{(\infty)}$ is *independent* of initial distribution $\pi^{(0)}$
  - $\pi^{(\infty)} = [1, 0, 0, \cdots, 0] \cdot P^\infty$ and thus the first row of P^∞ equals $\pi^{(\infty)}$
  - $\pi^{(\infty)} = [0, 1, 0, \cdots, 0] \cdot P^\infty$ and thus the second row of P^∞ equals $\pi^{(\infty)}$
  - etcetera

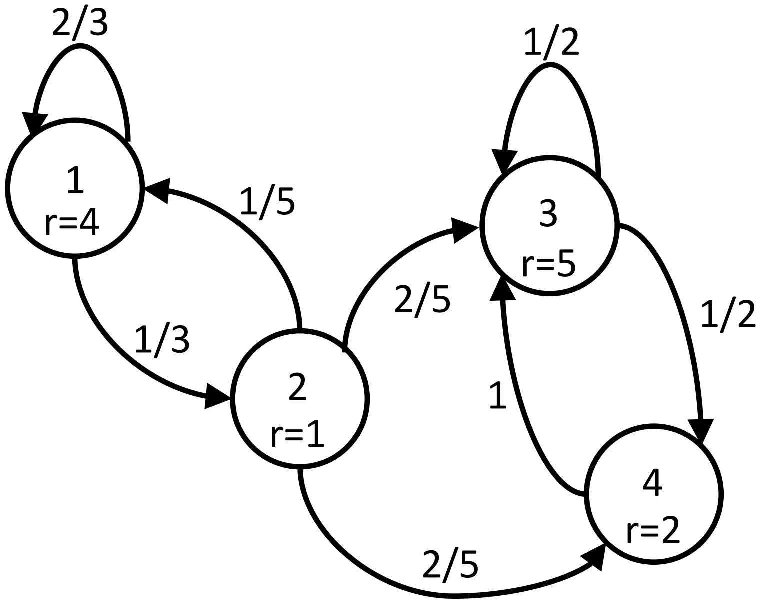
- P^∞ exists for every ergodic unichain, where each row equals $\pi^{(\infty)}$
  - if we know P^∞, $\pi^{(\infty)}$ can be computed by $\pi^{(\infty)} = \pi^{(0)} \cdot P^\infty$ (for any initial distribution)

ergodic unichain – recurrent versus transient states

- at some point in time transient states will not be visited anymore
  - $\pi_j^{(\infty)} = 0$ for any transient state j
  - $P_{\mathrm{ij}}^{\infty} = 0$ for any state i and transient state j

- if a recurrent state is visited once, it will be visited infinitely often
  - $\pi_j^{(\infty)} > 0$ for any recurrent state j
  - $P_{\mathrm{ij}}^{\infty} > 0$ for any state i and recurrent state j

ergodic unichain – example



$$P = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} & 0 & 0 \\ \frac{1}{5} & 0 & \frac{2}{5} & \frac{2}{5} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$P^{\infty} = \begin{bmatrix} 0 & 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & 0 & \frac{2}{3} & \frac{1}{3} \end{bmatrix}$$

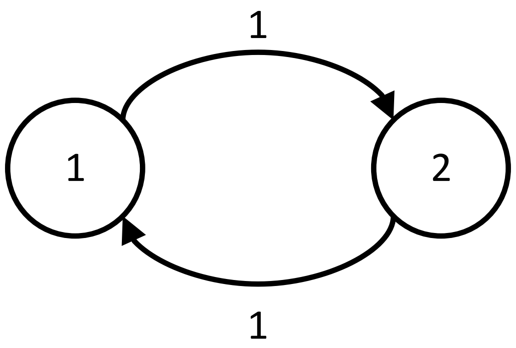
- Balance equations: $\pi = \pi \cdot P, \sum_{i \in S} \pi_i = 1$
  - $\pi_1 = \frac{2}{3}\pi_1 + \frac{1}{5}\pi_2$
  - $\pi_2 = \frac{1}{3}\pi_1$
  - $\pi_3 = \frac{2}{5}\pi_2 + \frac{1}{2}\pi_3 + \pi_4$
  - $\pi_4 = \frac{2}{5}\pi_2 + \frac{1}{2}\pi_3$
  - $\pi_1 + \pi_2 + \pi_3 + \pi_4 = 1$
- Solution $\pi^{(\infty)} = [0, 0, \frac{2}{3}, \frac{1}{3}]$
- long-run average reward: $\left[0, 0, \frac{2}{3}, \frac{1}{3}\right] \cdot [4,1,5,2]^T = 4$
- P_{ij}^{∞}: long-run probability that chain will be in j starting from i

long-run analysis – exercises

- Section B.6 in the course notes
  - Exercise B.28 (Limiting matrix ergodic unichain)
    - use CMBW (DTMC) to compute / verify answer
      1. create the model
      2. select ‘Determine MC Type’ to determine the type of Markov chain (to find out what limit is actually computed)
      3. select ‘Limiting Matrix’ to compute the limiting matrix
  - Exercise B.29 (Video application – limiting distribution)
    - use CMBW (DTMC) verify answer for different values of p
      1. create the model (the same model as Exercise B.24) for different transition probabilities
      2. select ‘Determine MC Type’ to determine the type of Markov chain
      3. select ‘Limiting Distribution’ to compute the limiting distribution
- answers are provided in Section B.8 of the course notes

non-ergodic unichain

- when unichain is non-ergodic, $\pi^{(\infty)}$, P^{∞} and $\pi^{(\infty)} r^T$ do not exist
  - e.g. assume $P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and $\pi^0 = [1, 0]$
  - then $P^1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $P^2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, $P^3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\cdots$
  - and $\pi^{(1)} = [0, 1]$, $\pi^{(2)} = [1, 0]$, $\pi^{(3)} = [0, 1]$, $\cdots$



- however, the balance equations of a (periodic) unichain still have a unique solution
  - balance equation have a unique solution iff the chain is a unichain

- what does this unique stationary distribution represent?

Cezàro limit

- the unique stationary distribution equals the so-called Cezàro limit of π^n
  - $\pi^{(\propto)} = \lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} \pi^{(k)}$

- further $\pi^{(\propto)} = \pi^{(0)} \cdot P^{\propto}$ where $P^{\propto}$ is the Cezàro limit of P^n
  - $P^{\propto} = \lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} P^k$
  - each row of $P^{\propto}$ equals $\pi^{(\propto)}$

- long-run expected *average* reward equals $\pi^{(\propto)} \cdot r^T$
  - $\pi^{(\propto)} \cdot r^T = \lim_{n\to\infty} E(\frac{1}{n} \sum_{k=0}^{n-1} r(X_k))$

Cezàro limit interpretation

- what is the interpretation of $\pi_i^{(\propto)}$?
  - note $\pi_i^{(\infty)}$ is the probability that in the long run the chain is in state i

- $\pi^{(\propto)} \cdot r^T = \lim_{n\to\infty} E(\frac{1}{n}\sum_{k=0}^{n-1} r(X_k))$ is the long-run expected average reward

- $\pi_i^{(\propto)}$ is obtained by taking $r^T = [0, \cdots, 0 , 1, 0, \cdots, 0]$
  - yields the long-run expected fraction of time the chain spends in state i

- similarly, $\mathrm{P}_{ij}^{\propto}$ is the long-run expected fraction of time the chains spends in state j, given that it started in i

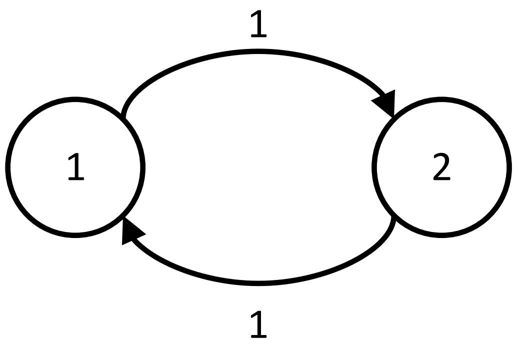
non-ergodic unichain – recurrent versus transient states

- at some point in time transient states will not be visited anymore
  - $\pi_j^{(\propto)} = 0$ for any transient state j
  - $P_{\text{ij}}^{\propto} = 0$ for any state i and transient state j

- if a recurrent state is visited once, it will be visited infinitely often
  - $\pi_j^{(\propto)} > 0$ for any recurrent state j
  - $P_{\text{ij}}^{\propto} > 0$ for any state i and recurrent state j

example

- $P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and $\pi^0 = [1, 0]$



- balance equation: $\pi_1 = \pi_2, \pi_2 = \pi_1, \pi_1 + \pi_2 = 1$ so $\pi_1 = \frac{1}{2}, \pi_2 = \frac{1}{2}$ and
  - $\pi^{(\propto)} = [\frac{1}{2}, \frac{1}{2}]$
  - $P^{\propto} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$
- the long-run expected fraction of time the chain spends in state 1 (or in state 2) equals $\frac{1}{2}$ (independent of the starting state)

long-run analysis – exercises

- Section B.6 in the lecture notes on Markov modeling, discrete event simulation
  - Exercise B.30 (Packet generator – generated load)
    - use CMBW (DTMC) to compute / verify answer
      1. create the model for different values of p
      2. select ‘Determine MC Type’ to determine the type of Markov chain (to find out what limit is actually computed)
      3. press ‘Limiting Distribution’ to compute the limiting distribution
  - Exercise B.31 (Expected fraction of time spent in a state equals reciprocal of expected return time)
    - use CMBW (DTMC) to verify answer
      1. create Markov reward model
      2. select ‘Determine MC Type’ to determine the type of Markov chain
      3. select ‘Limiting Distribution’ to compute the limiting distribution
      4. select ‘Reward until Hit’ to compute the expected return times
- answers are provided in Section B.8 of the course notes

general Markov chains

- balance equations of non-unichains do not have unique solution
  - limiting distributions *are dependent* of initial distribution

- the Cezàro limits $P^{\propto}$ and $\pi^{(\propto)}$ exist for any Markov chain
  - $\pi^{(\propto)} = \pi^{(0)} \cdot P^{\propto}$
  - long-run expected *average* reward equals $\pi^{(\propto)} \cdot r^T$

- the normal limits P^{∞} and $\pi^{(\infty)}$ exist if and only if the chain is ergodic
  - $\pi^{(\infty)} = \pi^{(0)} \cdot P^{\infty}$
  - long-run expected reward equals $\pi^{(\infty)} \cdot r^T$

computing the limiting matrices

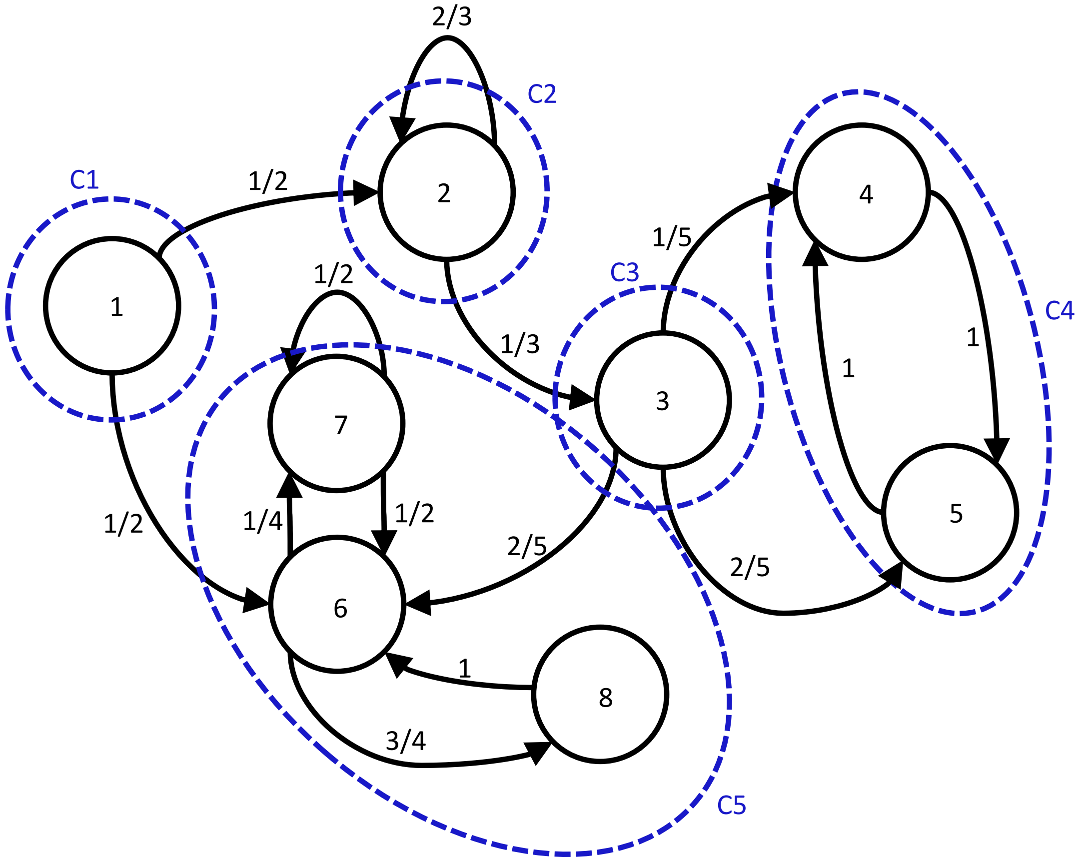
- what is $P_{ij}^{\propto}$?

- if state j is transient then $P_{ij}^{\propto} = 0$ for every state i

- if state j is recurrent it is part of some recurrent class C
  - starting from i, first end up in C: h_{iC} (hitting probability)
  - being captured in class C the long-run fraction of time spent in j equals $\pi_j^{(\propto)}$(the (unique) stationary solution to the balance equations of C)
  - so $P_{ij}^{\propto} = h_{iC} \cdot \pi_j^{(\propto)}$

- notice that $P_{ij}^{\infty} = P_{ij}^{\propto}$ in case P^{∞} exists

example



States 1, 2 and 3 are transient

Stationary distribution class C4

Stationary distribution class C5

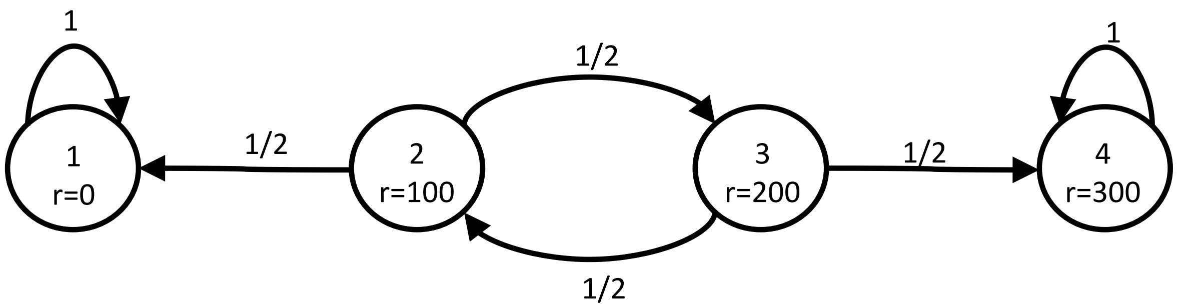
$P_{14} = h_{1C4} \cdot \pi_4^{C4} = 3/10 \cdot 1/2 = 3/20$

$P_{24} = h_{2C4} \cdot \pi_4^{C4} = 3/5 \cdot 1/2 = 3/10$

Etcetera

$$P^{\propto} = \begin{bmatrix} 0 & 0 & 0 & \frac{3}{20} & \frac{3}{20} & \frac{14}{45} & \frac{7}{45} & \frac{7}{30} \\ 0 & 0 & 0 & \frac{3}{10} & \frac{3}{10} & \frac{8}{45} & \frac{4}{45} & \frac{2}{15} \\ 0 & 0 & 0 & \frac{3}{10} & \frac{3}{10} & \frac{8}{45} & \frac{4}{45} & \frac{2}{15} \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{4}{9} & \frac{2}{9} & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 0 & \frac{4}{9} & \frac{2}{9} & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 0 & \frac{4}{9} & \frac{2}{9} & \frac{1}{3} \end{bmatrix}$$

example – gambler's ruin



$$P^{10} \approx \begin{bmatrix} 1.000 & 0.000 & 0.000 & 0.000 \\ 0.667 & 0.001 & 0.000 & 0.333 \\ 0.333 & 0.000 & 0.001 & 0.667 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix}$$

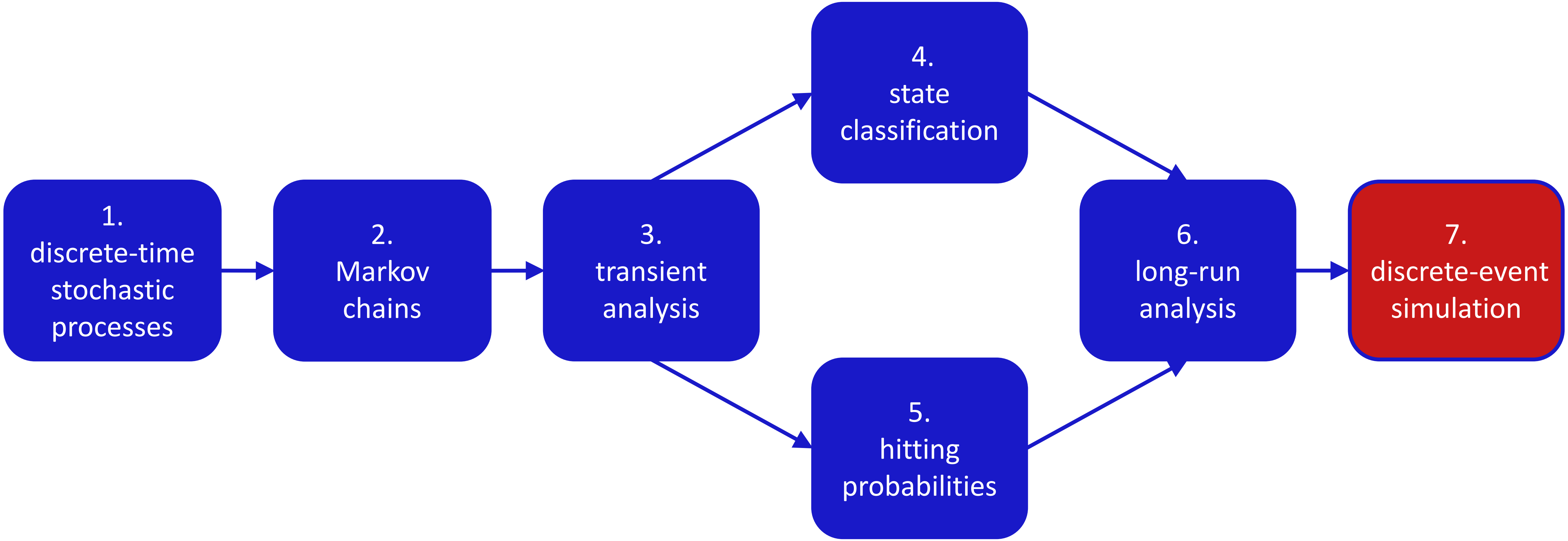
$$P^{\infty} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{2}{3} & 0 & 0 & \frac{1}{3} \\ \frac{1}{3} & 0 & 0 & \frac{2}{3} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- what is the long-run expected amount of cash for $\pi^{(0)} = [0, \frac{1}{5}, \frac{4}{5}, 0]$?
- answer:
  - $\pi^{(\infty)} \cdot r^T = \pi^{(0)} \cdot P^{\infty} \cdot r^T =$ $\left[\frac{6}{15}, 0, 0, \frac{9}{15}\right] \cdot [0, 100, 200, 300]^T = 180$ €

long-run analysis– exercises

- Section B.6 in the course notes
  - Example B.18 (Computing limiting matrix)
    - use CMWB (DTMC) to verify answer
      - create the model
      - select ‘Limiting Matrix’ to compute the Cezàro limiting matrix
  - Exercise B.32 (Composition of two parallel packet generators – generated load)
    - use CMWB (DTMC) to verify answer
      1. create Markov reward model for different values of p
      2. select ‘Determine MC Type’ to determine the type of Markov chain
      3. select ‘Limiting Distribution’ to compute the limiting distribution
      4. select ‘Long-run Reward’ to compute the long-run expected (average) reward
      5. select ‘Limiting Matrix’ to compute the (normal or Cezàro) limiting matrix
  - Exercise B.33 (Computer system – throughput)
    - use CMWB (DTMC) to verify answer
- answers are provided in Section B.8 of the course notes

module B - submodules and dependencies



lessons learned

Markov chain type	unichain	non-unichain
ergodic	(i) normal limits and Cezàro limits exist and are identical (ii) $\pi^{(\infty)}r^{T}$ is long-run expected reward (iii) $\pi^{(\infty)}$ is unique solution of balance equations; $\pi^{(\infty)}$ independent of $\pi^{(0)}$ (iv) all rows in P^{∞} are identical to $\pi^{(\infty)}$	(i) normal limits and Cezàro limits exist and are identical (ii) $\pi^{(\infty)}r^{T}$ is long-run expected reward (iii) balance equations have multiple solutions; $\pi^{(\infty)}$ is dependent on $\pi^{(0)}$ (iv) rows in P^{∞} are not all equal
non-ergodic	(i) only Cezàro limits exist (ii) $\pi^{(\propto)}r^{T}$ is long-run expected average reward (iii) $\pi^{(\propto)}$ is unique solution of balance equations; $\pi^{(\propto)}$ is independent of $\pi^{(0)}$ (iv) all rows in $P^{\propto}$ are identical to $\pi^{(\propto)}$	(i) only Cezàro limits exist (ii) $\pi^{(\propto)}r^{T}$ is long-run expected average reward (iii) balance equations have multiple solutions; $\pi^{(\propto)}$ is dependent on $\pi^{(0)}$ (iv) rows in $P^{\propto}$ are not all equal

questions

- how to compute limiting matrices?
- how to compute limiting distributions?
- how to compute limiting rewards?

questions

- how to compute limiting matrices? one can always compute the Cezàro limit by solving balance equations for the recurrent classes and use hitting probabilities; the normal limit only exists in case of an ergodic chain; in that case the normal limit and the Cezàro limit are equal
- how to compute limiting distributions? one can always compute the Cezàro limit using the initial distribution and the limiting matrix; the normal limit only exists in case of an ergodic chain; in that case the normal limit and the Cezàro limit are equal; in case of a unichain the limiting distribution equals the unique solution to the balance equations
- how to compute limiting rewards? one can always compute the Cezàro long-run expected average reward by multiplying the limit distribution with the transpose of the reward vector; in case of an ergodic chain this equals the long-run expected reward

questions

- how to estimate performance metrics in case computation is infeasible?

Markov modeling, discrete-event simulation – Module B

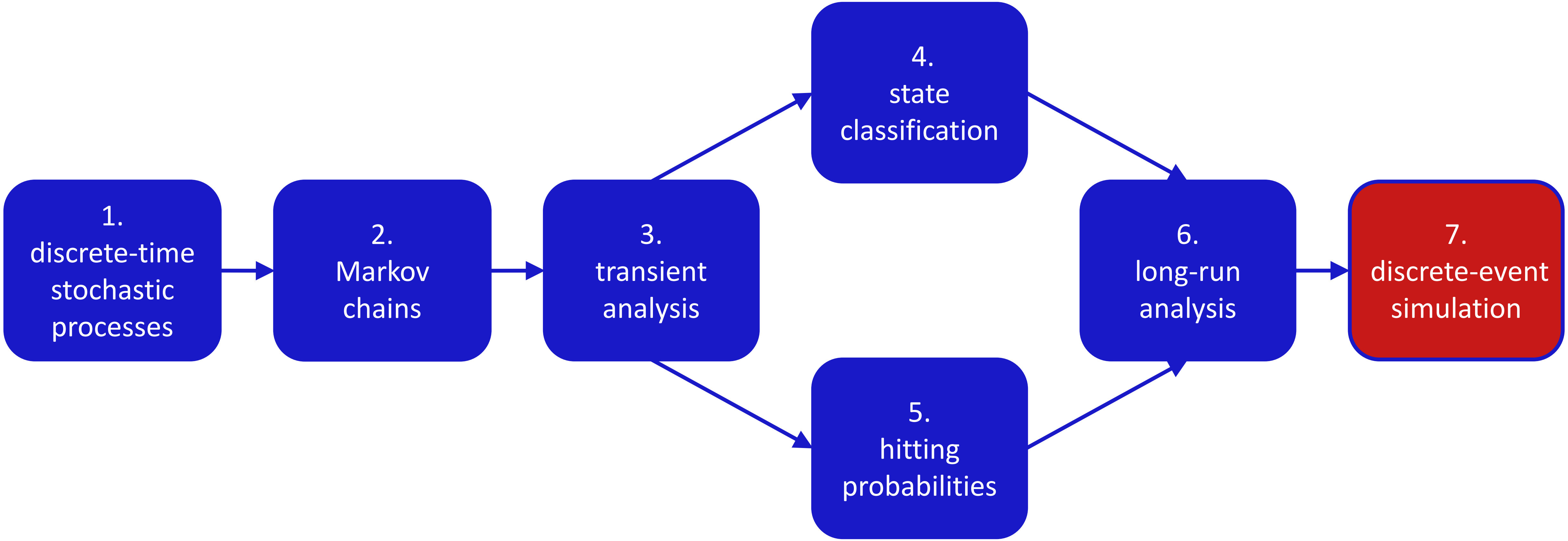
5XIE0 Computational Modeling

Twan Basten, Marc Geilen, Jeroen Voeten
Electronic Systems Group, Department of Electrical Engineering

credits and sources

- wikipedia:
  - https://en.wikipedia.org/wiki/Markov_chain
- Seminal textbooks on probability theory and Markov theory:
  - A.F. Karr. Probability. Springer, 1993.
  - N. Privault. Understanding Markov Chains, Examples and Applications. Springer, 2018.
  - J.R. Norris. Markov Chains. Cambridge University Press, 2012.
  - D.R. Cox and H.D. Miller. The Theory of Stochastic Processes. Springer, 1967.
  - K.L. Chung. Markov Chains with Stationary Transition Probabilities. Springer, 1967.

module B - submodules and dependencies



B.7 – discrete-event simulation

rational

- transition diagrams are rather primitive to model complex systems
  - offer only states and transitions
  - modeling two parallel packet generators (Exercise 32), requires manual definition of combined state space

- tools used in performance modeling practice typically offer convenient modeling constructs (such as parallel composition)
  - the more primitive Markov models are then still used, but as underlying semantic models that are either *implicitly* or *explicitly* defined
  - the state space of these underlying Markov models quickly grow very large or can even be infinite
    - e.g. 100 packet generators delivers a state space with 2^{100} states

discrete-event simulation

- so often Markov models are not explicitly available or are too large to *compute* transient or long-run properties

- instead of *computing* performance properties, tools often rely on **discrete-event simulation**
  - one or more paths are generated through the state space, starting in some initial state
  - upon the *event* of transitioning, a next state is chosen and visited
  - the choice should respect the transition probabilities and is made by a (pseudo)random generator

basic estimation theory

- consider sequence $Y_0, Y_1, \cdots$ of identically distributed independent random variables
  - $E(Y_n) = \mu$
  - $Var(Y_n) = E\left((Y_n - \mu)^2\right) = \sigma^2$

- goal: estimate μ based on a finite simulation sequence $y_0, y_1, \cdots, y_{M-1}$ of length M
  - each y_n is a realization of random variable Y_n

- approach: estimate μ by $\hat{\mu} = \frac{1}{M}\sum_{n=0}^{M-1} y_n$

- the fact that $\hat{\mu}$ is an estimation of μ is based on the **strong law of large numbers**

strong law of large numbers

- sequence $\frac{1}{M}\sum_{n=0}^{M-1} Y_n$ converges *almost surely* to the random variable constant μ which takes value μ with probability 1

- the *probability* of the set of sequences $y_0, y_1, \cdots$ for which $\lim_{M\to\infty} \frac{1}{M}\sum_{n=0}^{M-1} y_n = \mu$ equals 1
  - hence for any infinite sequence $y_0, y_1, \cdots$ we know *almost surely* that $\lim_{M\to\infty} \frac{1}{M}\sum_{n=0}^{M-1} y_n = \mu$
  - therefore $\hat{\mu} = \frac{1}{M}\sum_{n=0}^{M-1} y_n$ is approximately equal to μ

- value $\hat{\mu}$ is called a *point estimation* of μ
- random variable $\frac{1}{M}\sum_{n=0}^{M-1} Y_n$ is called a *point estimator* of μ

estimation errors

- $|\mu - \hat{\mu}|$ is the **absolute estimation error**
- $\frac{|\mu - \hat{\mu}|}{|\mu|}$ is the **relative estimation error**

- since $\lim_{M\to\infty} \frac{1}{M} \sum_{n=0}^{M-1} y_n = \mu$ (almost surely) we can make these errors as small as desired by choosing M sufficiently large

- so how long should the simulation sequence be so that we obtain acceptable errors?

- the foundation to answer this question is the **central limit theorem**

central limit theorem

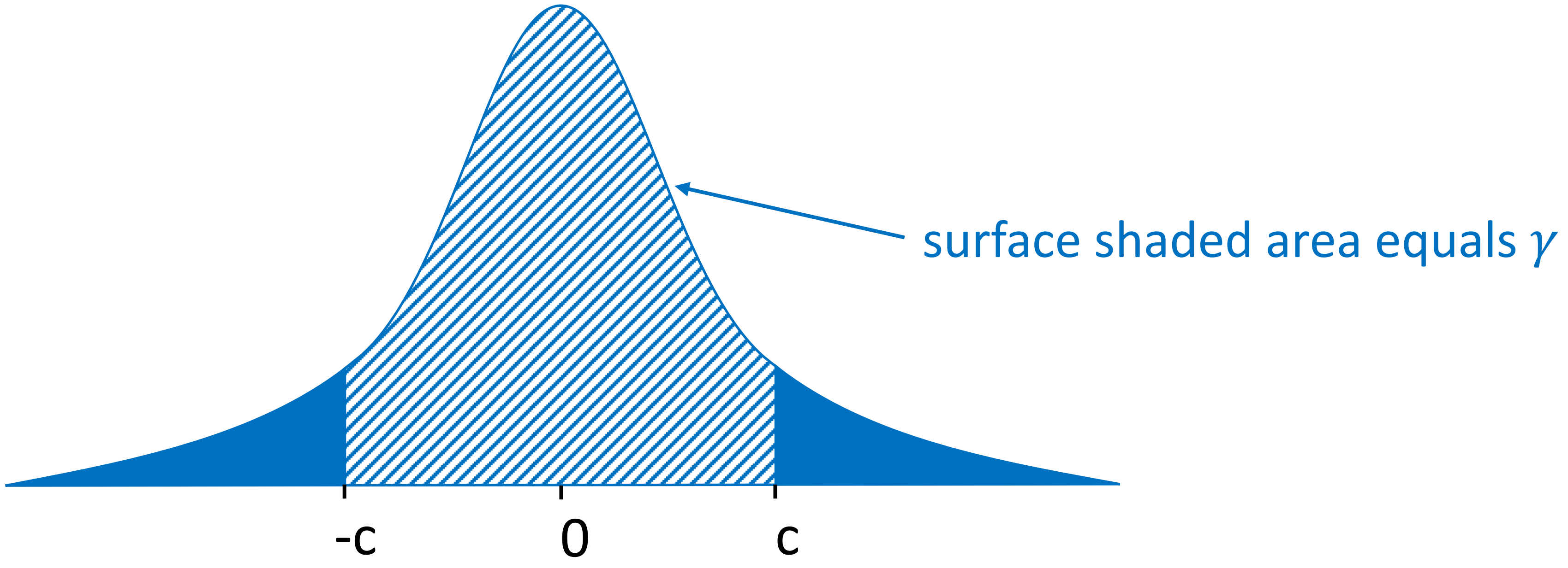
- $\sum_{n=0}^{M-1} Y_n$ has approximately a **normal distribution**
  - expected value: $M\mu$
  - variance: $M\sigma^2$
  - rule-of-thumb: $M \geq 30$

- the *normalized* variable $\sqrt{M}\frac{\frac{1}{M}\sum_{n=0}^{M-1} Y_n - \mu}{\sigma}$ then has approximately a *standard normal distribution*
  - *converges in distribution* to $N(0,1)$

- this also holds when *standard deviation* σ is replaced by point estimator S_M
  - $S_M = \sqrt{\frac{1}{M}\sum_{n=0}^{M-1}\left(Y_n - \frac{1}{M}\sum_{m=0}^{M-1} Y_m\right)^2}$ converges strongly to σ

confidence level

- choose **confidence level** γ and determine constant c such that $P(-c \leq N(0,1) \leq c)$



- thus $\mathrm{P}\left(-\mathrm{c} \leq \sqrt{M}\,\frac{\frac{1}{M}\sum_{n=0}^{M-1} Y_n - \mu}{S_M} \leq c\right) \approx \gamma$

interval estimator

- stochastic interval $[\frac{1}{M}\sum_{n=0}^{M-1} Y_n - \frac{cS_M}{\sqrt{M}}, \frac{1}{M}\sum_{n=0}^{M-1} Y_n + \frac{cS_M}{\sqrt{M}}]$ contains μ with probability γ (approximately)
  - **interval estimator** of μ

- about fraction γ of a large number K of realization (**interval estimations**) of this stochastic interval contains μ

realization/simulation 1 [$\widehat{\mu_1}$]

realization/simulation 2 [$\widehat{\mu_2}$]

realization/simulation 3 [$\widehat{\mu_3}$]

… …

realization/simulation K [$\widehat{\mu_K}$]

μ

confidence interval

- a realization $[\frac{1}{M}\sum_{n=0}^{M-1} y_n - \frac{cs_M}{\sqrt{M}}, \frac{1}{M}\sum_{n=0}^{M-1} y_n + \frac{cs_M}{\sqrt{M}}]$ is called a **confidence interval**
  - it is said that with confidence γ this interval contains μ
  - typical confidence levels used are 0.95 or 0.99

- Notice
  - $\hat{\mu} = \frac{1}{M}\sum_{n=0}^{M-1} y_n$ is the middle of the interval
  - the width of the interval is proportional to c (which depends on γ)
  - the width of the interval is proportion to standard deviation s_M
  - the width of the interval is inversely proportional to $\sqrt{M}$

estimation error bounds

- assume μ is in the confidence interval $[\hat{\mu} - \frac{cs_M}{\sqrt{M}}, \hat{\mu} + \frac{cs_M}{\sqrt{M}}]$, then
  - the absolute estimation $|\mu - \hat{\mu}|$ is bounded by $\frac{cs_M}{\sqrt{M}}$
  - the relative estimation error $\frac{|\mu - \hat{\mu}|}{|\mu|}$ is bounded by $\frac{cs_M/\sqrt{M}}{\hat{\mu} - cs_M/\sqrt{M}}$
    - if the interval only contains positive values
- the error bounds converge to 0 when $M \to \infty$
- how long should we simulate to obtain acceptable errors?
  - if we target a maximal error of ϵ we keep on increasing the simulation sequence until the computed bounds are $\leq \epsilon$

example – Ethernet simulation

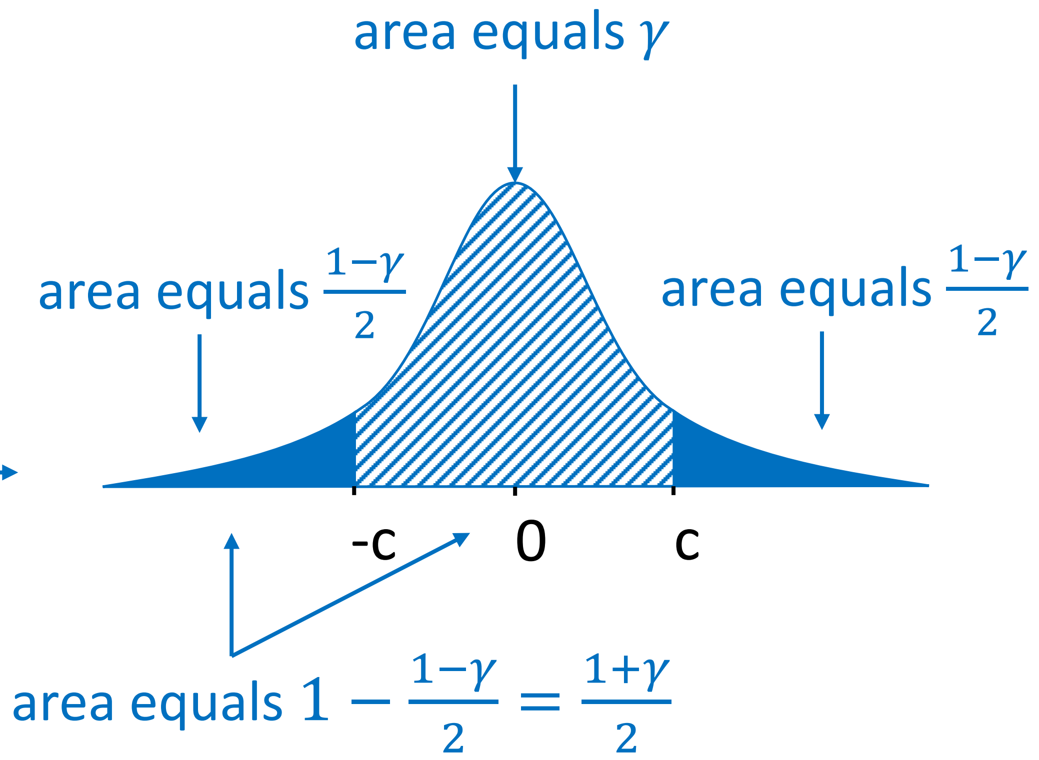
- consider slotted Ethernet behaving as sequence of independent Bernoulli-distributed variables $Y_0, Y_1, \cdots$
  - $P(Y_n = 1) = \mu$ (probability that a frame is sent in slot n)
  - $P(Y_n = 0) = 1 - \mu$ (probability that no frame is sent in slot n)
  - $E(Y_n) = 1 \cdot \mu + 0 \cdot (1 - \mu) = \mu$

- goal: estimate μ with confidence level $\gamma = 0.95$

- a simulation delivers the following sequence of 10 realizations ($M = 10$)
  - $y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9$ =1, 1, 0, 1, 0, 0, 1, 1, 0, 0

example – Ethernet simulation (cont'd)

- $\left[\hat{\mu} - \frac{c s_M}{\sqrt{M}}, \hat{\mu} + \frac{c s_M}{\sqrt{M}}\right] = [0.19,0.81]$
  - $\hat{\mu} = \frac{1}{M}\sum_{n=0}^{M-1} y_n \approx 0.5$
  - $s_M = \sqrt{\frac{1}{M}\sum_{n=0}^{M-1}\left(y_n - \frac{1}{M}\sum_{m=0}^{M-1} y_m\right)^2} \approx 0.5$
- determine c:
  - $P(-c \le N(0,1) \le c) = \gamma$ iff
  - $P(N(0,1) \le c) = \frac{1+\gamma}{2}$ iff
  - $F_{normal}(c) = \frac{1+\gamma}{2}$ iff
  - $F_{normal}^{-1}\left(\frac{1+\gamma}{2}\right) = c \approx 1.96$



example – Ethernet simulation (cont'd)

- error bounds
  - absolute: $\frac{cs_M}{\sqrt{M}} \approx 0.31$
  - relative: $\frac{cs_M/\sqrt{M}}{\hat{\mu}-cs_M/\sqrt{M}} \approx 1.6$ (=160%)

- longer simulation is required to get acceptable errors

discrete-event simulation – exercises

- Section B.7 in the course notes
  - Exercise B.34 (Throughput of an Ethernetwork – confidence levels versus error bounds)
  - Exercise B.35 (Throughput of an Ethernetwork - required length of simulation sequence)
  - Exercise B.36 (Interval estimator of expected value)
  - Exercise B.37 (Confidence interval interpretation)
  - Exercise B.38 (Standard deviation - point estimator)
- answers are provided in Section B.8 of the course notes

application to Markov chains – transient properties

1. define sequence of independent identically-distributed variables $Y_0, Y_1, \cdots$ such that $E(Y_n)$ equals property of interest
2. define how to obtain the observations y_n (based on generating paths through the Markov chain)
3. point and interval estimations and error bounds follow from basic theory

example – expected transient reward

- goal: estimate $E(r(X_M))$ for some fixed time value M

1. define each Y_n as an *independent copy* of $r(X_M)$
   - then (obviously) $E(Y_n) = E(r(X_M))$
2. realization $y_n = r(i_M)$ where
   - i_M is the final state of path $i_0 i_1 \cdots i_M$ of length M through the Markov chain
   - i_0 is randomly chosen based on $\pi^{(0)}$
   - state transitions are randomly chosen based on P
3. compute estimations (such as $\frac{1}{M}\sum_{n=0}^{M-1} y_n$)

discrete-event simulation – exercises

- section B.7 in the course notes
  - Exercise B.39 (Estimation transient distributions)
  - Exercise B.40 (Gambler's ruin – expected reward estimation)
    - use CMBW (DTMC) to compute / verify answer
      1. create the model and compute $\pi^{(2)}$ and $E(r(X_2))$
      2. select 'Transient Reward' in 'Simulation-based Operations on Markov Chains' and enter 2 as the number of steps. As stopping criteria use a confidence level of 95% and 10,710 as the maximum number of paths. The other input fields can be left unspecified.
  - Exercise B.41 (Gambler's ruin – converge rate central limit theorem)
    - use CMBW (DTMC) to compute / verify answer
      1. create the model (same as Exercise B.40) and for (a) compute $\pi^{(10)}$
      2. for (b) select 'Transient Distribution' in 'Simulation-based Operations on Markov Chains' and enter 10 as the number of steps. As stopping criteria use a confidence level of 95% and use 10,000 as the maximum number of paths. The other input fields are left unspecified.
      3. for (c) change the rewards assigned to the states, select 'Transient Reward' and use as stopping criteria an absolute error bound of 0.0001 and 200,000 as the maximum number of paths. The other input fields are left unspecified. To obtain more accurate estimations, leave the absolute error bound unspecified.

discrete-event simulation – exercises

- section B.7 in the course notes
  - Exercise B.42 (Estimation hitting probability – impact of maximal path length)
    - use CMBW (DTMC) to compute / verify answer
      1. create the model (same as Example B.18)
      2. select ‘Hitting Probability’ (in ‘Simulation-based Operations on Markov Chains’) and select state S3. As stopping criteria use a maximal path length of 3 and use 71734 as the maximum number of paths. The other fields are left unspecified.
  - Exercise B.43 (Estimation expected cumulative reward until hit)
  - Exercise B.44 (Rover in a maze – estimation escape probability)
    - use CMBW (DTMC) to compute / verify answer
      1. create the model (same as Exercise B.27)
      2. select ‘Hitting Probability Set’ and select state S29. As stopping criteria use a relative error bound of 0.01, specify the maximum number of paths to be 1000,000, and use different maximum path lengths. The other fields are left unspecified.
- answers are provided in Section B.8 of the course notes

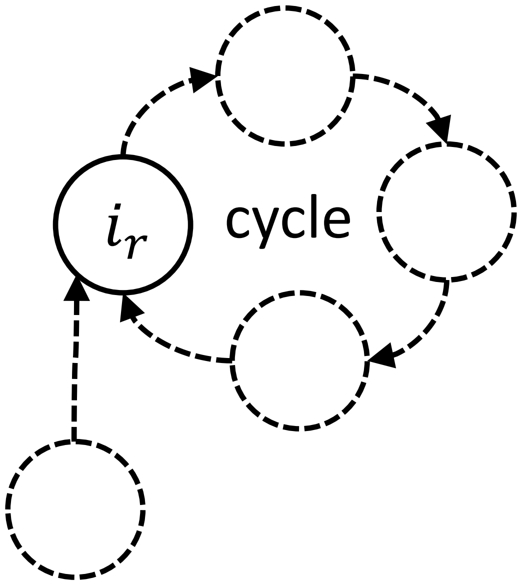
application to Markov chains – long-run properties

- estimation of long-run properties is hard for Markov chains in general
- consider special case of *long-run expected average rewards* $\pi^{(\propto)} \cdot r^T$ for *unichains*, denoted by μ

1. define point estimator/estimation for μ based on strong law of large numbers and sequence of appropriate random variables
2. define interval estimator/estimation for μ based on central limit theorem
3. define how to obtain sequence of realizations based on paths in unichain

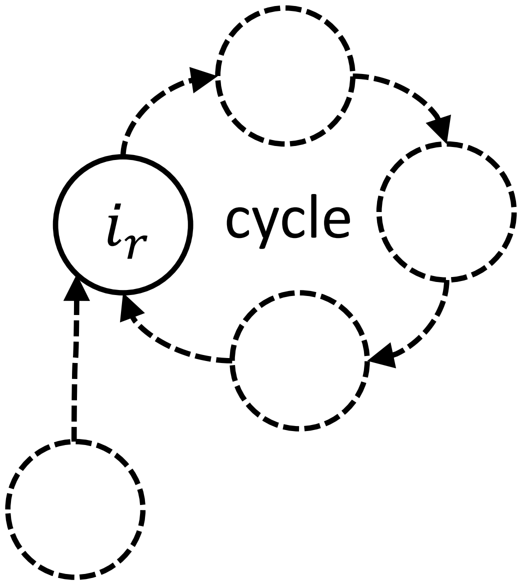
long-run expected average reward – point estimation (1)

- observation: any unichain has recurrent state i_r
  - that will be visited from any state with probability 1
  - that will be revisited with probability 1
  - through which the unichain keeps on cycling



- define variable L_n as length of n^{th} cycle through i_r, then
  - $L_0, L_1, \cdots$ are independent identically-distributed
  - $E(L_n) = \frac{1}{\pi_{i_r}^{(\propto)}}$ (see Exercise 31)
  - $\frac{1}{M}\sum_{n=0}^{M-1} L_n$ converges almost surely to $\frac{1}{\pi_{i_r}^{(\propto)}}$ and is a point estimator thereof

long-run expected average reward – point estimation (1)



- define variable R_n as cumulative reward obtain in the n^{th} cycle through i_r, then
  - $R_0, R_1, \cdots$ are independent identically-distributed
  - $E(R_n) = f^r_{i_r i_r} = \frac{\mu}{\pi^{(\infty)}_{i_r}}$ (see Exercise 42)
  - $\frac{1}{M}\sum_{n=0}^{M-1} R_n$ converges almost surely to $\frac{\mu}{\pi^{(\infty)}_{i_r}}$ and is a point estimator thereof
- hence $\frac{\sum_{n=0}^{M-1} R\_n}{\sum_{n=0}^{M-1} L\_n}$ converges almost surely to $\frac{\mu/\pi^{(\infty)}_{i_r}}{1/\pi^{(\infty)}_{i_r}}$ = μ and is a point estimator thereof
- $\hat{\mu}$ = $\frac{\sum_{n=0}^{M-1} r\_n}{\sum_{n=0}^{M-1} l\_n}$ is a point estimation of μ

long-run expected average reward – interval estimation (2)

- define variable Y_n as $R_n - \mu L_n$
  - $Y_0, Y_1, \cdots$ are independent identically-distributed
  - $E(Y_n) = \frac{\mu}{\pi_{i_r}^{(\infty)}} - \mu \frac{1}{\pi_{i_r}^{(\infty)}} = 0$
  - assume $var(Y_n) = \sigma^2$
- then by the central limit theorem $\frac{\sum_{n=0}^{M-1} Y\_n}{\sqrt{M}\sigma}$ has approximately a standard normal distribution
  - hence $\mathrm{P}\left(-\mathrm{c} \leq \frac{\sum_{n=0}^{M-1} Y_n}{\sqrt{M}\sigma} \leq c\right) = \gamma$ (where c=$F_{normal}^{-1}\left(\frac{1+\gamma}{2}\right)$)
  - this also holds when σ is replaced by $S_M = \sqrt{\frac{1}{M}\sum_{n=0}^{M-1}\left(R_n - \frac{\sum_{m=0}^{M-1} R_m}{\sum_{m=0}^{M-1} L_m}\right)^2}$

long-run expected average reward – interval estimation (2)

- interval estimator for μ
  - $\left[\frac{\sum_{n=0}^{M-1} R\_n}{\sum_{n=0}^{M-1} L\_n} - \frac{cS_M}{\sqrt{M}\frac{1}{M}\sum_{n=0}^{M-1} L_n}, \frac{\sum_{n=0}^{M-1} R_n}{\sum_{n=0}^{M-1} L_n} + \frac{cS_M}{\sqrt{M}\frac{1}{M}\sum_{n=0}^{M-1} L_n}\right]$
- confidence interval
  - $\left[\hat{\mu} - \frac{cS_M}{\sqrt{M}\frac{1}{M}\sum_{n=0}^{M-1} l_n}, \hat{\mu} + \frac{cS_M}{\sqrt{M}\frac{1}{M}\sum_{n=0}^{M-1} l_n}\right]$
- error bounds are derived in similar way as before

long-run expected average reward – generate observations (3)

- generate single long path through the unichain
  - initial segment (from chosen start state until i_r is hit for first time) is ignored
  - l_n and r_n are based on segment of path between n plus first and n plus second hit of i_r

- in a practical simulation choosing a recurrent state or determining whether a state is (re)visited is often not feasible
  - start next recurrency cycle based on the value of a reward (e.g. indicating that a queue has run empty)
  - start next segment if number of states in current segment reaches a predefined threshold – *technique of batching*

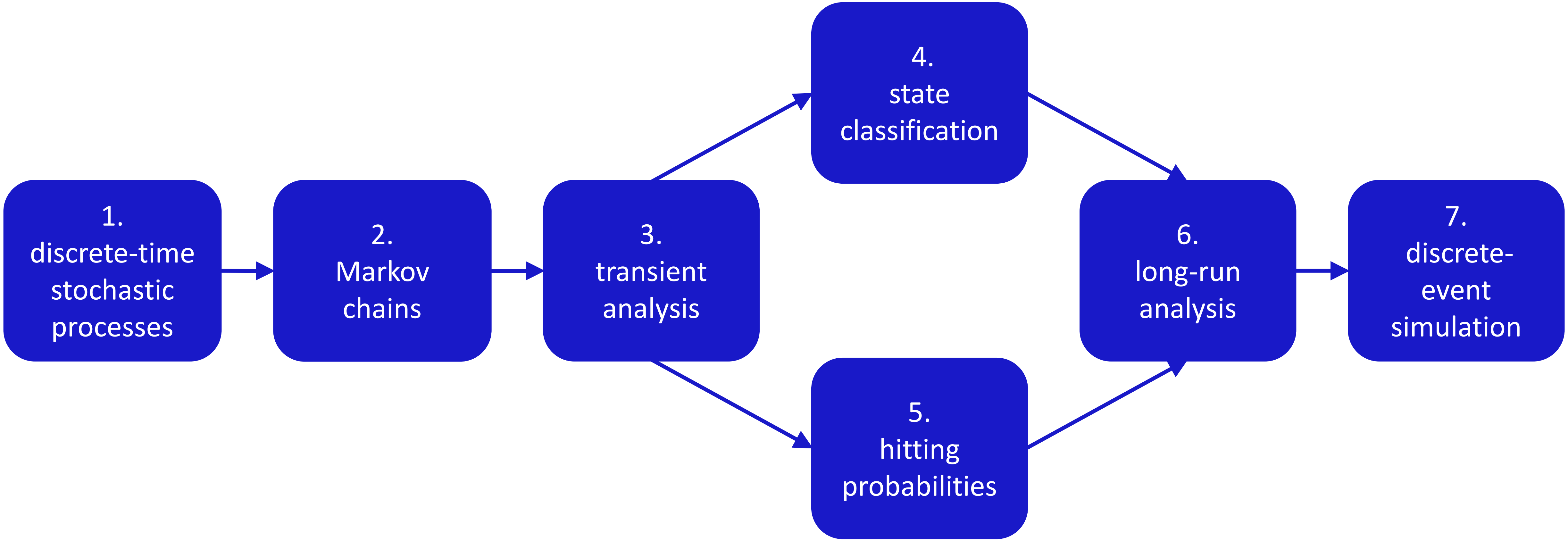
discrete-event simulation – exercises

- section B.7 in the course notes
  - Exercise B.45 (Absolute error bound long-run expected average reward)
  - Exercise B.46 (Long-run expected average reward – confidence interval)
  - Exercise B.47 (Expected reward until return versus long-run expected average reward)
    - use CMBW (DTMC) to compute / verify answer
  - Exercise B.48 (Interval estimator long-run expected average reward)
  - Exercise B.49 (Estimation long-run expected fraction of time spent in a state)
- answers are provided in Section B.8 of the course notes

discrete-event simulation – exercises

- section B.7 in the course notes
  - Exercise B.50 (Estimation Cezàro limiting distribution of a non-unichain)
    - use CMBW (DTMC) to compute / verify answer
      1. create the model (same as Example B.18)
      2. select 'Cezàro Limit Distribution' and choose 'No preference' regarding the recurrent state. Use the default stopping criteria.
  - Exercise B.51 (Video application – long-run average buffer occupancy)
    - use CMBW (DTMC) to compute / verify answer
      1. create the model (same as Exercise B.24) and specify the rewards
      2. select 'Long-run Average Reward' and choose appropriate stopping criteria
- answers are provided in Section B.8 of the course notes

module B - submodules and dependencies



lesson learned

- **point estimators** of expected values are based on **strong law of large numbers**
- a realization of a point estimator is called a **point estimation**
- **interval estimations** of expected values are based on the **central limit theorem**
- a realization of an interval estimator is called an **interval estimation**
- a **confidence interval** is an interval estimation that contains the value to estimate with a certain **confidence level**
- **absolute** and **relative error bounds** can be derived from confidence intervals; they form a stop criterium for the simulation

questions

- how to estimate performance metrics in case computation is infeasible? determining performance metrics in performance practice is often based on discrete-event simulation